Information Work Support Based on Activity Data

Handlungsdatenbasierte Informationsarbeitsunterstützung
Dissertation zur Erlangung des Grades eines Doktor-Ingenieurs (Dr.-Ing.)
Eingereich von Dipl.-Inform., Dipl.-Medienwiss. Benedikt Schmidt,
geboren am 06.12.1981 in Frankenberg

Angenommen vom Fachbereich Informatik der Technischen Universität Darmstadt

1. Gutachten: Prof. Dr. Max Mühlhäuser
2. Gutachten: Prof. Dr. Albrecht Schmidt

Tag der Einreichung: 14.05.2013
Tag der Disputation: 27.06.2013
Darmstadt 2013—D17

TECHNISCHE UNIVERSITÄT DARMSTADT

Telecooperation Lab at
Technische Universität Darmstadt

Man is the symbol-using (symbol-making, symbol-misusing) animal,
inventor of the negative (or moralized by the negative),
separated from his natural condition by instruments of his own making,
goaded by the spirit of hierarchy (or moved by the sense of order),
and rotten with perfection.

Burke. Language as Symbolic Action. [39]

Information Work Support Based on Activity Data
Handlungsdatenbasierte Informationsarbeitsunterstützung

Eingereich von Dipl.-Inform., Dipl.-Medienwiss. Benedikt Schmidt,
geboren am 06.12.1981 in Frankenberg

Angenommen vom Fachbereich Informatik der Technischen Universität Darmstadt

1. Gutachten: Prof. Dr. Max Mühlhäuser
2. Gutachten: Prof. Dr. Albrecht Schmidt

Tag der Einreichung: 14.05.2013
Tag der Disputation: 27.06.2013

D17—Darmstadt 2013

Ehrenwörtliche Erklärung[2]

Hiermit erkläre ich, die vorgelegte Arbeit zur Erlangung des akademischen Grades Dr.-Ing. mit dem Titel ”Information Work Support Based on Activity Data“ selbständig und ausschließlich unter Verwendung der angegebenen Hilfsmittel erstellt zu haben. Ich habe bisher noch keinen Promotionsversuch unternommen.

Darmstadt, den 14.05.2013

Dipl. Inform., Dipl. Medienwiss. Benedikt Schmidt

[2] Gemäß §9 Abs. 1 der Promotionsordnung der TU Darmstadt

Abstract

In industrial and post industrial nations like Germany and the USA more than a quarter of the workforce mainly works with information. Most of the work done by these information workers is the production, supervision and dissemination of information at computer workplaces. Information workers frequently works on multiple tasks in parallel. Few guidelines regulate and structure the work process. Therefore, the successful execution of the work requires a high degree of individual planning.

A common effect of ad-hoc executions of multiple tasks are memory failures: Planned activities are forgotten (prospective memory failures), or the recall of work processes' status and involved information objects fails (retrospective memory failures). The computer—a multitasking machine—even increases the likelihood of memory failures due to an increased number of activities executed in parallel.

This dissertation investigates methods to decrease the likelihood of memory failures in information work at the computer workplace. The effort leads to the design of a tool that provides support for information work based on externalized activity data. This document is structured as follows:

- The first part investigates information work from the perspectives of psychology, organization theory and sociology. Identified characteristics of information work relevant for this dissertation are captured in an ideal type. This includes the specification of the information work process at the computer workplace as being coordinated by interruptions and as being composed of logical units of work, so called knowledge actions and desktop operations.

- The second part proposes a system design method which facilitates the analysis of work processes that can be typically observed in information work. The method seamlessly integrates into the user-centred design method. Work is modeled and analyzed in terms of so called activity system models based on activity theory and action regulation theory. System model and analysis realize two important elements of the user-centred design method: the context of use analysis and the requirement specification. The specified method is applied to the domain of information work, resulting in requirements for a tool to decrease the likelihood of memory failures in information work.

- The third part develops methods to address memory failures in information work based on activity data. The developed methods address the requirements previously identified by applying the system design method (part 2) to the identified ideal type (part 1). The methods are implemented and evaluated in a demonstrator:

  - Activity Data: A fundamental contribution to address memory failures is the collection of information about the work process. To realize this, methods to capture, analyze and organize interaction histories are developed. A core element of the process is activity mining, which is a method to identify activities in interaction histories even if the activities were interrupted during the execution process. Activity mining is modeled as a clustering problem. The proposed activity mining methods show better results than the state of the art with respect to the identification of activities. Furthermore, the proposed activity mining methods extract more details about the work execution process than the state of the art.
  - Methods to Address Memory Failures: Based on the extracted activity data the goal of this work is realized—support methods to address memory failures at the computer workplace are developed. A support method design space to address memory failures is created. The design space is structured along three support directions (exploration, organization, recommendation). For each support direction, a respective user support method has been designed: 1) Activity-centric task management, which leverages activity data to facilitate task management and to support the recall of ongoing activities and respective work processes. 2) An interactive activity history, which enables the exploration of activity data in a work history visualization to support the recall of earlier work processes. 3) A recommender system, which analyzes the most recent work activities of the user to propose useful information objects like emails, files and websites. The system can be configured to support for more multitasking oriented or for more focused work.
  - Transparency Tool: The support methods have been implemented in a demonstrator named Transparency. Using the demonstrator an evaluation of the support methods with a focus on memory support was conducted. The evaluation results indicate that the support methods decrease the likelihood of prospective and retrospective memory failures for information work at the computer workplace.

The scientific contributions of this dissertation address two domains. On the one hand, information work support. Methods are developed which decrease the likelihood of prospective and retrospective memory failures based on activity data. On the other hand, system design methods. A method is introduced to design systems for work types which involve a high degree of individual planning.

Kurzfassung (deutsch)

Mehr als ein Viertel der Arbeitnehmer in (Post-)Industrienationen wie Deutschland oder den USA arbeiten mit Daten. Ein Großteil der eigentlichen Arbeitsleistung dieser Informationsarbeiter entfällt auf die Produktion, Überwachung und Verteilung von Information am Computer Arbeitsplatz. Informationsarbeiter bearbeiten meist zeitgleich verschiedene Aufgaben während der eigentliche Arbeitsprozess durch wenige Auflagen strukturiert wird. Aus diesem Grund benötigt die erfolgreiche Ausführung der Arbeit stetige Koordination und Planung.

Die Vielzahl gleichzeitiger Aufgaben und die Komplexität des Arbeitsprozesses im Zusammenhang mit Unterbrechungen löst Erinnerungsfehler aus: Informationsarbeiter vergessen geplante Aktivitäten (planungsbezogene Erinnerungsfehler), den Status von Arbeit und die genutzten Informationsobjekte (ereignisbezogene Erinnerungsfehler). Der Computer als Multitasking Maschine steigert die Wahrscheinlichkeit von Erinnerungsfehlern sogar noch insofern mehr Aufgaben gleichzeitig durchgeführt werden.

Diese Dissertation untersucht Methoden zur Minderung von Erinnerungsfehlern in der Informationsarbeit am Computerarbeitsplatz. Im Rahmen der Arbeit wird eine Anwendung zur Unterstützung von Informationsarbeit auf Grundlage von externalisierten Aktivitätsdaten entwickelt. Die Arbeit ist wie folgt strukturiert:

- Der erste Teil untersucht Informationsarbeit aus der Perspektive der Psychologie, der Organisationstheorie und der Soziologie. So identifizierte Charakteristiken der Informationsarbeit die im Kontext dieser Arbeit wichtig sind werden in einem Ideal Typ abgebildet. Dies beinhaltet die Beschreibung der Informationsarbeit am Computerarbeitsplatz als koordiniert durch Unterbrechungen und als zusammengesetzt aus logischen Arbeitseinheiten, so genannten Knowledge Actions und Desktop Operations.

- Der zweite Teil beschreibt eine System Design Methode welche die Analyse von Arbeitsprozessen wie sie in der Informationsarbeit auftreten erleichtert. Die beschriebene Methode erweitert die user-centred design Methode. Aufbauend auf Activity Theory und Handlungsregulationstheorie wird Arbeit mittels sogenannter Aktivitätssystem Modelle abgebildet und analysiert. Abbildung und Analyse setzen zwei wichtige Elemente der user-centred design Methode um: die Erhebung des Nutzungskontextes und die Anforderungsanalyse. Die entwickelte Methode wird in der Dissertation verwendet, um Anforderungen an eine Software zur Minderung von Erinnerungsfehlern in der Informationsarbeit abzuleiten.

- Im dritten Teil werden Methoden zur Adressierung von Erinnerungsfehlern unter Nutzung von Aktivitätsdaten beschrieben. Dabei werden die auf Grundlage des Ideal Typs (Teil 1) und mittels System Design Methode identifizierten Anforderungen (Teil 2) adressiert. Die Methoden werden in einem Demonstrator umgesetzt und evaluiert:

  – Aktivitätsdaten: Einen grundlegenden Beitrag zur Adressierung von Erinnerungsfehlern können Informationen über den Arbeitsprozess leisten. Um diese Daten bereitstellen zu können werden Methoden zur Erhebung von Interaktionshistorien am Computer sowie deren Analyse und Organisation beschrieben.

    Wesentlicher Bestandteil des Vorgehens ist Activity Mining, ein Verfahren, um Aktivitäten in Interaktionshistorien zu identifizieren auch wenn die Aktivitäten im Arbeitsprozess unterbrochen worden sind. Activity Mining wird als Clustering Problem beschrieben. Die beschriebenen Ansätze zeigen bessere Resultate beim Identifizieren von Aktivitäten als der Stand der Forschung. Weiterhin sind die extrahierten Informationen über den Arbeitsprozess reichhaltiger als beim Stand der Forschung.

  – Methoden zur Adressierung von Erinnerungsfehlern: Auf Grundlage der extrahierten Aktivitätsdaten wird das grundlegende Ziel der Arbeit realisiert—Unterstützungsmethoden zur Adressierung von Erinnerungsfehlern bei der Informationsarbeit am Computer-arbeitsplatz werden entwickelt. Richtlinien für die Entwicklung von Unterstützungsmethoden die Erinnerungsfehler adressieren werden identifiziert und resultieren in der Entwicklung von drei wesentlichen Beiträgen: 1) Activity-centric Task Management (aktivitätsbezogene Aufgaben Verwaltung) nutzt Aktivitätsdaten, um Aufgabenverwaltung zu erleichtern und unterstützt so das Erinnern an laufende Aufgaben und zugehörige Arbeitsprozesse. 2) Eine Interactive Activity History (interaktive Aufgabenhistorie) unterstützt das Erinnern an Arbeitsprozesse durch die Exploration einer Arbeitsprozess Visualisierung. 3) Ein Recommender System (Vorschlagssystem) schlägt Informationsobjekte (Emails, Dateien, Webseiten, etc.) auf Grundlage der vorhergehenden Handlungen des Informationsarbeiters vor. Das System kann so konfiguriert werden, dass ein eher Multitasking orientiertes oder ein eher fokussiertes Arbeiten unterstützt wird.

  – Transparency Tool: Die entwickelten Methoden sind in einem Demonstrator namens Transparency prototypisch implementiert worden. Mittels Transparency sind die Methoden im Hinblick auf ihre Erinnerungsunterstützung mit

Informationsarbeitern evaluiert worden. Die Ergebnisse legen nahe, dass die entwickelten Methoden die Wahrscheinlichkeit planungsbezogener und ereignisbezogener Erinnerungsfehler in der Informationsarbeit am Computerarbeitsplatz senken.

Zusammenfassend betreffen die wissenschaftlichen Beiträge dieser Arbeit zwei Themenbereiche. Die Beiträge betreffen einerseits die Unterstützung von Informationsarbeit. Es werden Methoden entwickelt, welche die Wahrscheinlich von planungsbezogenen und ereignisbezogenen Erinnerungsfehlern mittels Aktivitätsdaten verringern. Die Beiträge betreffen andererseits die Entwicklung von System Design Methoden. Eine Methode wird vorgestellt, die insbesondere das System Design für Arbeiten mit hoher Autonomie und Planung erleichtert.

Acknowledgement

This dissertation would not have been possible without the support of colleagues, friends and family.

I want to thank Prof. Dr. Max Mühlhäuser for supporting this work and for offering advice and help during the last 4 years. Max, thank you! Furthermore, I thank Prof. Dr. Albrecht Schmidt for being my co-referee.

Activity theory is an important topic within this dissertation. Sebastian Döweling, thank you for many discussions resulting in a myriad of diagrams. Uwe Riss, thank you for introducing me to activity theory and to the insight that knowledge is no commodity but a rational capacity.

Different colleagues have supported my work. In particular I want to thank Markus Döhring, Andreas Faatz, Eicke Godehardt, Andreas Göb, Francesco Novelli, Heiko Paulheim, Markus Schief, and Axel Schulz.

Wolfgang Reinhardt, thank you for the collaboration on knowledge actions and the related discussions. Robin Marterer, it has been a pleasure to discuss the nature of activities in different domains.

The development of the prototypes described in this thesis was supported by different students. In particular I want to thank Johannes Boppre, Johannes Kastl and Björn Pantel.

Special thanks go to my managers, Knut Manske and Bettina Laugwitz. Knut, thank you for the support and trust over the years. Bettina, thank you for the support with respect to empirical studies and especially for introducing the work of Hacker to me.

Furthermore, thanks go to my project coordinators during the time. Nicolas Liebau, thank you for many good advices and for helping me to focus on this dissertation. Todor Stoitsev, thank you for the first joint year at SAP Research, the support during paper writing and the discussions.

I want to thank my friends, in particular the Paderborner Schule.

Last but not least, I am deeply grateful for the support of my family, my parents as well as Christian, Nicola, Christin-Sophie and Leonie-Alexandra.

Contents

1 Introduction

In industrial and post-industrial nations like Germany and the USA more than a quarter of the workforce works with information [47, 213, 220]. The relevance of this type of work has steadily increased since Machlup first described it in the 1960s [168, 213]. Most of the work done by these information workers is the production, supervision and dissemination of information at computer workplaces. They play a major role in the coordination and control of today's economy—an economy organized in networks of commodity and information exchange on a global scale [25].

The computer workplace is the major place to unfold information worker productivity, consuming up to 50 percent of the work day [190, 101]. Information workers access roughly 170 information objects daily (incl. 90 websites, 73 emails [137]) which are created, accessed and modified in work processes composed of WIMP[1] style interactions with operating systems and a variety of applications [279].

The information worker performs non-routine work, multitasks, decides what needs to be done, what to do next and how to do it, while considering constraints and limitations. As an effect, the information worker is prone to work on a multitude of tasks in parallel and has to identify appropriate work processes to finalize the tasks successfully. Interruptions play a crucial role for the work process. On the one hand self-interruptions are an important mechanism of work coordination, used by the individual to thoughtfully switch between activities while on the other hand, external interruptions may enforce unplanned task switches [180, 232].

In summary, the information worker workforce has a steadily increasing relevance. The work done by this workforce is shaped to a high degree by autonomy which results in multitasking and ad-hoc work processes.

1.1 Motivation

Due to the high degree of autonomy the information worker faces the crucial challenge of self organization [138, 1]: Information workers handle a multitude of parallel goals which are realized in complex work processes. This type of work has a severe drawback: it often causes memory failures. Information workers are prone to forget planned goals (prospective memory failures), work processes' status and involved information objects (retrospective memory failures) [67, 37]. Memory failures result in tedious work activities like duplicated search efforts or even forgotten work items and consequently have a negative effect on the efficiency of the information worker [240].

Computers are an important tool for the information worker. Since the first computers have been used in business contexts, the machines have evolved to a multitasking enabled, elaborate communication technology which provides access to a myriad of information. As an effect, computers empower the individual to increase efficiency. Nevertheless, the enabled parallelism of work and the amount of accessible information generate constant intentional self distractions [227] and even increases the likelihood of prospective and retrospective memory failures.

In conclusion the modern computer workplace has increased the likelihood of memory failures during work execution. Thus, the question is whether the likelihood of memory failures can be decreased by appropriate software. Self organization in order to avoid memory failures has especially been addressed in the domain of personal information management [24, 48]: personal information management includes the externalization of goals, respective activities and information objects which help the information worker to remember relevant facts of the work process. Most tools for personal information management have negative effects on the work process by 1) causing additional distraction due to required data maintenance activities [223] or by changing the way information workers work in an unwise manner like completely avoiding interruptions [201]. Other approaches focus on the unobtrusive collection of data about the work process to support information work [216, 163] by recommending information objects in specific work situations.

Research on user support is an ongoing challenge because memory failures remain a relevant threat for information workers. Therefore, the key motivating question for this thesis is: *How to decrease memory failures involved in information work at the computer workplace?*

In the remainder of this chapter, a set of objectives is identified that needs to be realized to answer that question (see section 1.2). Respective challenges are identified and the methodology as well as a scope for the work is set (see sections 1.3, 1.4 and 1.5). Finally, the structure and the contributions of this dissertation are provided (see sections 1.6). The connection between these elements is visible in Figure 1.1.

[1] Abbrv.: Windows, Icons, Menus, Pointers

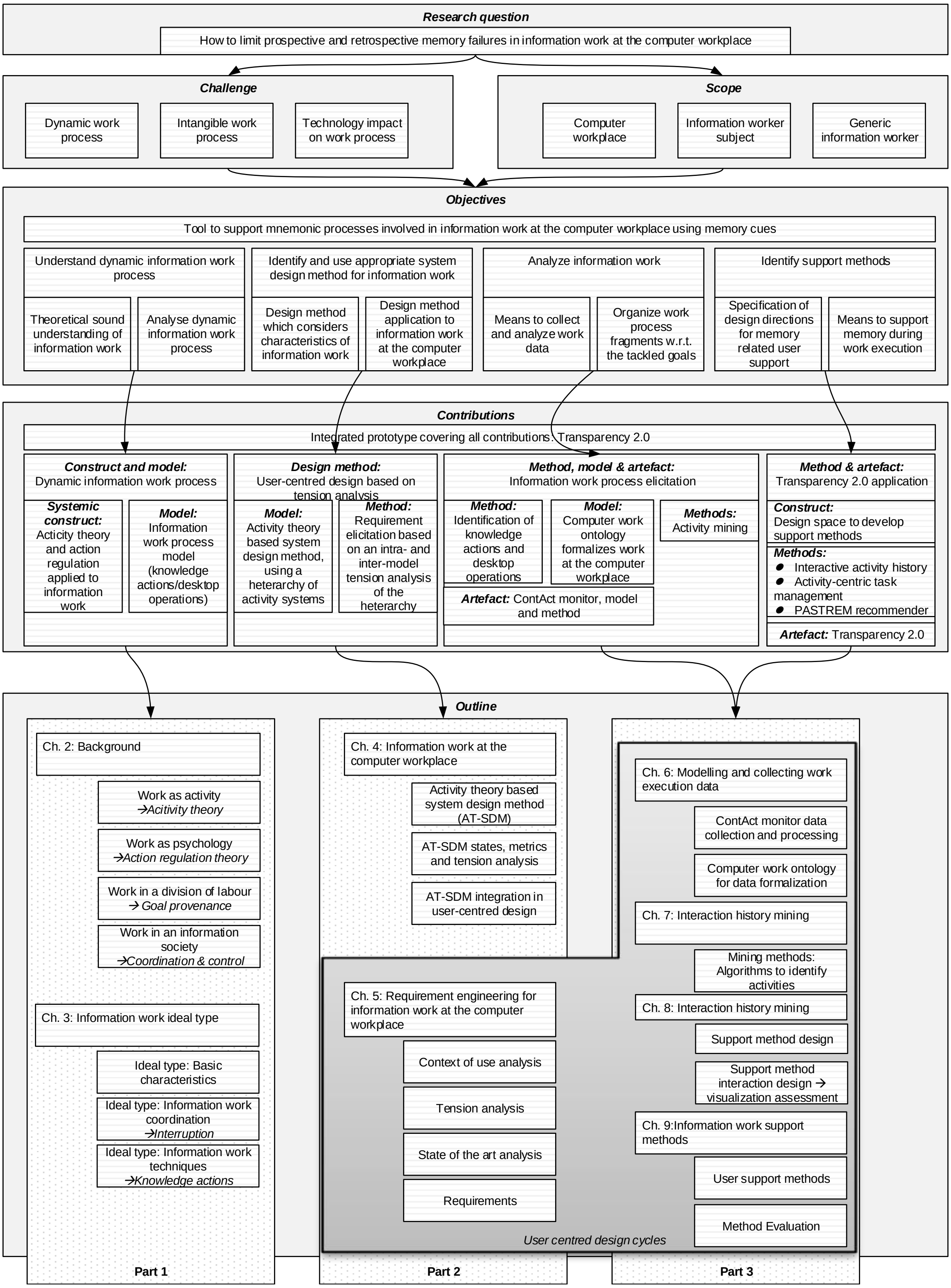

Figure 1.1.: Connection between the research question, related objectives, contributions and respective chapters of this dissertation.

1.2 Objectives

The goal of this dissertation is to *decrease the likelihood of memory failures in information work at the computer workplace*. In order to limit memory threats of the information worker the recall processes of the subject need to be supported. A basic mechanism to support recall processes is the use of memory cues [277] (also referred to as memory triggers [37] or memory prosthesis [150]). A cue increases recall likelihood, i.e., it helps to remember things quicker and with more details (cf. [13]), like a picture which helps to recall the episode of events happening while the picture was taken. Considering the example of the picture it is obvious that different types of externalized information can serve as memory cue. In the following, activity data will be used as a memory cue for the information worker. To make this approach plausible and to specify actual user support methods, background about work execution is needed. The background knowledge will facilitate the specification of a system design for information work support which results in the design and the implementation of a system:

1. ***Background***

   To prepare system design for information work support, a decent understanding of information work is required (challenge 1).

   a) Objective: Conduct a systematic review of research about information work.

   b) Objective: Identify an approach to model the information work process, especially with respect to its coordination and the logical units of work it consists of.

2. ***System design method***

   The gained understanding of information work needs to provide requirements for software to address memory failures.

   a) Objective: Identify a system design method appropriate for information work.

   b) Objective: Elicit requirements for an information work support tool, using the identified system design method.

3. ***Design and Implementation***

   Based on the requirements memory cues should be identified to be used for a software to address memory threats. This includes:

   a) Objective: Identify means to create memory cues based on collected data.

   b) Objective: Identify means to offer memory cues to the information worker.

4. ***Evaluation***

   a) Objective: Finally, an evaluation based on user experiments needs to proof that the created software actually decreases the likelihood of memory failures.

1.3 Challenges

Information work has different characteristics which complicate the realization of the aforementioned objectives. These challenges are described in the following:

1.3.1 Challenge 1: Autonomy

Although information workers are part of controlled organizational structures and need to follow many predefined processes, their work execution is largely autonomous. Autonomy means that the information worker decides by himself which goals to pursue in which manner under consideration of constraints. A goal is pursued by executing activities. The multitude of goals force the information worker to organize the time spent with activities to realize goals. This results in frequent switches between different activities.

Example: Document authoring is a mixture of activities that is split into subtasks by interruptions. The information worker has the goal of creating a document. This goal triggers a document authoring activity. The information worker authors the document with a word processor, realized by basic operations of mouse clicks and mouse movements. During the authoring process an email notification appears. The information worker switches to the email program as his goal of being informed about new information has a high priority. He reads the email, replies and goes back to the word processor to continue the authoring. Few minutes later, the information worker realizes an implication of the information in the email related to his work. Thus, another work goal generates a new activity which results in another activity switch: The information worker interrupts the authoring to contact a colleague to verify the implications of the email.

Effect of challenge: The dynamic work process complicates the analysis of information work to derive the requirements for a support system. Generally, system design methods support the design of systems for a specific goal. The methods describe means of analyzing the goal specific work process and of eliciting requirements to intentionally modify the work process. A design method to address information work execution needs to consider the relevance of decisions for activities and the resulting activity switches.

1.3.2 Challenge 2: Intangible Work Process

To pursue goals, information workers realize activities which require an interaction with the world. For the computer workplace, activities are composed of basic operations like mouse movements and clicks. Facts which stand for interactions with information based on visual interfaces. However, it is not obvious how the relation between the interaction with the world and the underlying intent is structured. The reason of a mouse movement is not obvious.

Example: During document authoring the information worker formulates sentences in his mind, writes them down, deletes words and adds new words. While an observer only recognizes a stream of keyboard inputs, the underlying decisions to drop and add words remain unknown.

Effect of challenge: The intangible nature of information work needs to be addressed. Models for information work need to be created and methods need to be identified to gain an understanding of the interplay between cognitive activities and the interaction with the world. This understanding is required to reason about memory threats.

1.3.3 Challenge 3: Technology Impact on the Work Process

Introducing software affects the way people work and thus creates new work practices. Therefore, the design of technology needs to consider the willingness of the prospective users to integrate the technology into their work process. Additionally, the prospective effects of the technology on the work processes it interferes with need to be considered. For information work this aspect is of specific importance due to the dynamic and the intangible nature of work.

Example: An information work support tool collects data about information worker goals in order of providing support. The data is collected by frequently asking the information worker "What is your goal?' (e.g., Clippy, the office assistant which resulted from research on user need anticipation [125]). The data collection is an interruption on its own which negatively affects the work process.

Effect of challenge: The impact of a solution needs to be considered. Schultze stresses that one needs to observe *what the doing does* [249]: "practices need to be understood in the context of their circuits of reproduction, i.e., the reciprocal, cyclical relationships through which practice creates and recreates the objectified social structures and the conditions in which it occurs" [249]. To address this, not only a solid understanding of information work is required but also a system design phase which transfers the understanding of information work into a system design while closely investigating effects of the tool on the work process.

1.4 Research Methodology

The dissertation follows a design science approach. In contrast to behavioral science which develops theories to explain phenomena, design science is an engineering like approach to deliver practical solutions to attain goals [118]. Thus, the coordinated transformation of phenomena, following their inherent causalities and relations based on a design, possibly manifested in an artifact is intended [260].

Design science follows explanatory, predictive and normative theory as prescriptive statements and methods are identified, useful to be manifested in artifacts. Design science results are constructs, models, methods and artifacts. The artifact as implementation is the highest order result of design science [118]. Empirical foundation is used to show the validity of constructs, models, methods and artifacts based on observations.

Design science is closely related to system design methods which have a specific focus on artifact design and implementation. This dissertation implements design science with a specific consideration of system design methods. First constructs and models for information work are identified and serve as input for user-centered design (UCD) as system design method. UCD is a system design method which incorporates the user in all phases of the software development process to achieve a usable system [198, 136, 171]. The UCD is an iterative design solution which is composed of four fundamental processes: context of use analysis, requirement elicitation, system design and system evaluation. Methods exist to address the processes, e.g., task analysis for the context of use analysis. Here, UCD transforms the constructs and models of information work into a support solution for mnemonic processes involved in information work.

To sum up, in this dissertation design science is the overall methodology which is implemented, using UCD.

1.5 Research Scope

The scope of this thesis is constrained by the following aspects:

- **Constraint 1: The Computer Workplace**
  The focus of this thesis is the computer workplace of the information worker. The computer workplace is an important source of memory threats (which will be shown) and covers a relevant fragment of the information worker's workday (50 % of the workday is spent at the computer [190, 101]).

- **Constraint 2: The Information Worker Subject**
  Within this dissertation only information worker individuals are considered. This does not mean that a solipsist perspective on the information worker is nurtured. The social environment including aspects like collaboration and hierarchical connectedness is considered implicitly because it affects the individual's work process.

- **Constraint 3: The Generic Information Work**
  The dissertation analyzes information work as a generic type of work with a set of basic characteristics. A deeper classification of different types of information work is avoided. This helps to provide a general understanding of information work and provides a foundation for specialization in future research.

1.6 Outline

This dissertation covers a process from analyzing information work to system design, design implementation and evaluation of applications to address memory threats. The process can be roughly separated into three parts:

- ***Part I – Information work foundations***

  - The *Background* chapter 2 introduces theories and background information on work used throughout the whole dissertation. The chapter presents work from the domains of psychology, organization theory and sociology. Work psychology specifies work as goal directed interaction of a subject with the world in terms of activities based on activity-theory (AT) and action regulation theory (ART). The organization theory perspective specifies work as being delegated in a division of labor. The sociology perspective provides an explanation of specific types of work based on conditions and requirements of societal formations. Work is identified as a product of society which requires an understanding of the environment. In this sense, the information worker is introduced as product of societal conditions related to globalization and a crisis of control at the end of the 20th century.
  - Chapter 3 specifies an ideal type for information work execution at the computer workplace. The ideal type is a unified analytical construct to reason about memory threats during information work execution. It specifies the coordination of work execution based on complex cognitive processes in reference to the work psychology perspective delivered in the first chapter. First, *information work process coordination* is discussed based on a review of literature on *interruption*. It is shown that interruptions, despite their coordinative function, are a major source of memory failures. Second, an evaluation identifies logical units of work an information work process is composed: Knowledge actions as work techniques and desktop operations as basic interactions.

    Parts of the ideal type and the empirical research have been published in [245, 221, 247, 248, 162].

- ***Part II – System design for information work***

  - The delivery of a *system design method for information work* to be used within this dissertation is the main topic of chapter 4. An overview of existing system design methods is given. UCD is chosen as an appropriate design method. To address multiple goals and to incorporate the cognitive processes of work coordination within the design process, a system design method is developed based on concepts from AT and ART. The method is called activity theory based system design method (AT-SDM). It specifies a tension based analysis of a context of use modeled in terms of activity systems. The method seamlessly integrates into the UCD process.

    The basic characteristics of the AT-SDM have been published in [76, 237].
  - Chapter 5 applies UCD with AT-SDM to the domain of information work. The chapter focuses on the context of use analysis and requirement specification. The understanding gained on information work (chapter 2 and 3) is transferred into requirements for a tool to limit the likelihood of memory failures in information work at the computer workplace. *The basic idea of the identified requirements is to unobtrusively collect work execution data and to offer this data to the information worker to support recall processes.*

- ***Part III – Support tool development*** The third part of the dissertation addresses the identified requirements. On the one hand, methods of collecting and processing work process data, so called interaction histories, are introduced. On the other hand, the use of collected data to provide support to decrease the likelihood of memory failures is discussed.

- The *modelling and collection of work execution data* is presented in chapter 6. A process to enrich interaction histories with external knowledge and to identify logical units of work is provided. The method externalizes work execution data based on software sensors and enriches the data based on existing facts and heuristics to identify knowledge actions and desktop operations. The resulting activity data is formalized in an ontology named *computer work ontology (CWO)*, extending the DOLCE upper ontology [96]. Both contributions, method and model, are implemented in *the ContAct monitor, an artifact* which produces computer work ontology instances based on user observations.

  The contributions have been published in [242, 238].

- *Activity mining* presented in chapter 7 discusses and evaluates algorithms to identify activities within interaction histories. The investigated methods build on the interaction histories provided by the ContAct monitor, using the CWO ontology. Three different directions for activity mining are investigated, namely semantic approaches that focus on semantic similarity, process based approaches that focus on the graph structure of knowledge actions and a hybrid approach that combines both mentioned types. The methods are evaluated 1) against a gold standard and 2) during a long term study with several information workers.

  Parts of the contribution have been published in [241].

- Chapter 8 specifies a system design space for support methods based on interaction data. The design space specifies necessities and decisions involved in the development of activity data based user support methods. This bridges a gap between the identified requirements and the support method development: the requirements only specify information needs while the system design space specifies how to address these needs. The basic principle of the design space is to foster support methods which mediate the recall process of a subject.

- Based on the system design space, user support methods are developed in chapter 9. The support methods are in detail:

  * Activity-centric task management (organization): *Activity-centric task management* provides an overview of a subject's tasks. Activity data facilitates the creation and maintenance of the task objects and is used to provide additional work process information.

  * Interactive activity history (exploration): The *interactive activity history* gives access a work history. The subject's history is explorable based on a browser with filter and search capabilities.

  * PASTREM activity centric recommender (recommendation): The *PASTREM activity centric recommender* generates proactive recommendations of information objects based on the most recent work process of a subject.

  All support methods have been realized and integrated into an application with the name Transparency 2.0. Transparency 2.0 has been evaluated in a user study with a focus on memory failures. The study proofs that the created support methods decrease the likelihood of memory failures for information work at the computer workplace. This also shows the usefulness of activity data for user support and the usefulness of the AT-SDM as extension of UCD.

  The support methods have been described in [240, 239, 236, 237].

Part I.

Information Work

2 Background

The goal of this dissertation is to identify methods that limit the likelihood of memory failures in information work at the computer workplace. Therefore, an analysis of information work is required that unfolds the origins of memory failures. To approach information work, it is necessary to step back and to address the broader topic of work in general. This chapter provides the required understanding by delivering a theoretical foundation and prepares the analysis of information work (which is conducted in chapter 3).

Only based on an understanding of work in general the specific characteristics of information work emerge. Work as "activity in which one exerts strength or faculties to do or perform something" [184] obviously is no simple concept. In fact, the concept of work is subject of a variety of scientific disciplines. An observation of the spectrum helps to gain an understanding of the concept. This is provided in the following. Vocabulary and theories from three scientific disciplines are reported, namely cognitive psychology, organization theory and sociology. The three perspectives share different boundary points which help to acquire a broad perspective on work. The selection does not strive for completeness but has been chosen with respect to its relevance within the context of information work analysis:

- **Psychological perspective (see sections 2.1 and 2.2):** The psychological perspective (especially sociocultural psychology and work psychology) provides explanations how subjects execute work. The specific challenge of an execution perspective is to describe the interplay between a subject's cognitive processes and the actual interaction with the world. Even simple examples show the complexity: Considering chopping a tree opens many questions. To name only a few: How does the subject coordinate the action?, What is the result of a tool choice between a saw or an axe?, Which things are consciously coordinated and which happen almost automatically? Answers to these questions are provided by sociocultural psychology. One domain of sociocultural psychology is the analysis of mental processes involved in work execution [304]. Here, activity-theory (AT) and action regulation theory (ART) are presented which specify work as a goal directed activity (see section 2.1). Basically, the coordination of the cognitive processes and the actual tool based interaction between a subject and the real world in a productive process is specified. The gained perspective on work execution is of central importance for the information work analysis and for all methods developed within this thesis.

- **Organization theory perspective (see section 2.3):** Organization theory investigates into the effective, efficient and practical design of organizations [127]. The perspective provides explanation of work execution in organizational settings. The subject is no longer origin of the performed activities. In fact, the subject makes contract based commitments to accept delegated activities. The contract specifies the degree of specialization and the autonomy of the subject with respect to the work execution structure. Considering these aspects provides a good understanding of work in a market economy. This is relevant as the type of work considered in this thesis is situated in organizations structured according to the principles of the market economy.

- **Sociology perspective (see section 2.4):** Within sociology work is part of the social system. Thus, work has a relevance within the system and holds different relations to other elements of the social system. This helps to address an important question with respect to work: Why do certain types of work emerge? The answer to this question must have its roots in the social system which created the type of work. For this thesis: Why did information work emerge at the end of the twentieth century? Based on analytical and empirical work conducted by Beniger [25] and Castells [47] the emergence of work based on historical circumstances like economics and technological conditions is explained. Specific attention is given to information work. Evidence is provided that information work emerged at the end of the twentieth century to address requirements of a global economy. Based on new ways of information distribution and autonomy a crisis of control within the global economy was addressed. Having these conditions in mind, the analysis of information work is simplified.

The list of addressed perspectives shows that this chapter starts with a generic understanding of work and finally focuses information work with the description of the information society as social structure information work originated from.

2.1 Psychology I: Work as Activity in Activity Theory

AT describes the goal directed interaction of a subject with the world in terms of activities. Within the domain of human computer interaction, AT has gained increasing relevance as alternative to the dominating approach of cognitive psychology [141, 149, 144]. In contrast to cognitive psychology, AT does not only consider the interaction between a subject and a device but also considers the interaction context. Applications of AT within human computer interaction exist with respect to the context of use analysis and the system design [76].

In the following, an understanding of work as being composed of activities is conveyed based on AT. Activities which: 1) structure the interaction between a human being and the world, 2) mediate the interaction process based on tools, 3) are source of the subject's development. Additionally, AT is a framework which can be used to analyze activities based on activity systems and the hierarchical decomposition of activities. The section only provides those aspects of AT which are relevant for this dissertation. A fundamental and complete treatment of AT can be found, e.g., in the original works by Leontiev and Vygotsky [152, 293], as well as in newer introductory texts like [149, 140, 144]. Shorter introductions are also available in numerous publications on AT-informed designs, e.g., [91].

The section is structured as follows. First, the basic concepts of AT are provided (see section 2.1.1) to introduce the concept of activity. Second, the organization of activities within a hierarchical structure is provided (see section 2.1.2). Third, activity systems are introduced to illustrate the organization of the concepts and consider the system-based analysis of activities (see section 2.1.3).

2.1.1 Basic Concepts

The AT framework is an outgrowth of the sociocultural perspective of Russian psychology of the early 20th century [152, 153]. The sociocultural perspective expresses human development as a product of the world. In other words, the human mind is shaped by the generative forces of culture and society. This is a specific characteristic of AT as given factors like physiology are not accounted exclusively for human development capabilities.

To account for human development as product of the world, AT provides the concept of activity. The activity is the overarching concept of AT which situates and generates the human being as subject in a world of objects. Activities stand for interactions between a subject and an object. Objects compose the subject's environment and are not limited to physical entities but include social and cultural entities as well (e.g., a door, a language system as well as the concept of democracy is an object). Activities are distinguished based on their objects. Two activities are different if they have a different object [152].

The interaction between a subject and an object within an activity refers to a complex transformation process. At first glance, the subject transforms the object within an activity (e.g., a tree is chopped). In fact, the interaction covers two types of transformations as the subject is transformed by the activity, as well (e.g., the subject's chop skills increase). Therefore, activities, subjects and objects mutually determine each other. In this sense, subjects do not express themselves in their activities but subjects are produced by their activities (cf. [229]): "S $\longleftrightarrow$ O" [144]. The activity exposes itself as a "unit of life" [140] producing object and subject. Based on the relation between subject, object and activity as "unity of consciousness and activity" [141] AT accounts for human development as a product of the world.

In the following, the concepts mediation and internalization are introduced. These concepts provide a better understanding of how the interaction between subject and object within an activity is structured (based on mediation) and how the subject develops based on activities (based on internalization).

2.1.1.1 Concept of Mediation

Mediation refers to the mode of interaction between a subject and an object within an activity. The interaction is mediated by objects which are used as mediators—also referred to as tool—within an activity. Mediators accumulate and carry cultural practice—habits and modes of operation—which are reproduced and strengthened by mediator application. In the following, the term tool is used in favor of mediator for the sake of readability.

A tool designs the interaction with respect to the transformation process, its anticipation and its perception (e.g., chopping a tree with a saw differs from chopping a tree with an axe on a planning on an execution as well as on the perception level). Therefore, the tool choice has crucial relevance for an activity. Once a tool is chosen, the respective activity is organized based on the modes of operation suggested by the tool (e.g., a saw is designed to be utilized by a specific mode of operation). Therefore, the tool has a dichotomic position within activities: it belongs to the subject like an extension of the body while it is an object on its own. The subject faces the dichotomy and is able to switch between both positions within two levels of activity design. The first level of activity design addresses the tool choice. The second level of activity design accepts a tool choice and designs the activity based on the tool.

Tools as objects used to mediate the interaction process within an activity can be very different due to the broad object concept of AT (comprising physical, cognitive and social entities as well). In this respect, tools range from simple things like a stone, to complex constructs like language or algebra. Vygotsky proposed an organization of tools in two groups [293]:

- **Material tools which extend physical capabilities.** Examples for material tools are knives, levers, but may also be microscopes.

- **Psychological tools which extend mental abilities.** Psychological tools comprise "various systems for counting; mnemonic techniques; algebraic symbol systems; works of art; writing; schemes, diagrams, maps, and technical drawings; all sorts of conventional signs, and so on." [292] For Vygotsky "[the] sign acts as an instrument of psychological activity in a manner analogous to the role of a tool in labor" [293, p.52].

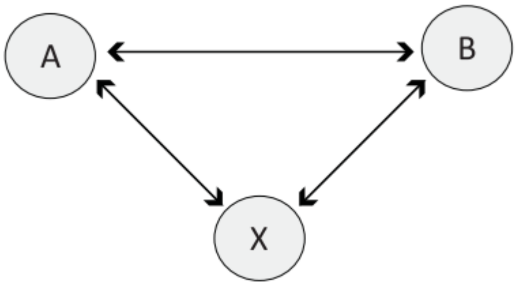

Figure 2.1.: The structure of an instrumental act, based on [294]. 'A—B' represents a simple association between two stimuli, underlying a natural mnemonic act. When memory transforms into a high-level psychological function, this association is replaced with an instrumental act comprising 'A—X' and 'X—B' taken from [144].

One must be particularly careful with respect to the manifestation of the two tool groups. At first glance, one may assume that material tools have a physical manifestation while psychological tools only have a mental representation. However, there is no direct relation between the manifestation and the groups. While material tools in general will, in fact, have a physical manifestation, psychological tools often have different representations, as a map can be printed on paper while it may also be only a mental representation.

2.1.1.2 Concept of Internalization

Internalization refers to the acquisition of mental capabilities based on interaction with the world. The acquisition process is a fundamental precondition for mediation. An object can only become a mediating tool if the subject is able to reproduce at least a mode of operation within an activity. Therefore, the subject applies internalization. Two different types of internalization can be distinguished:

- **Internalization of the mode of operation:** Internalization can be limited to the acquisition of a mode of operation (e.g., learning how to operate a saw to chop a tree).

- **Internalization of the object:** The internalization of the object itself is a complex type of internalization which is sometimes applicable. The subject abandons the external object representation and completely relies on an internal representation of the tool. If it is applicable, this type of internalization follows the internalization of the mode of operation. Object internalization frequently occurs for psychological tools (e.g., the subject first learns the modes of operations of a physical map while later on the map becomes part of the mental capabilities of the subject. The subject "knows" how the map is organized). Studies have shown that the internalization of objects generally results in an improved activity execution performance (less time, higher quality) [167]. The way internalization is realized is sketched in Figure 2.1. An association between two objects is transformed based on a higher mental function which replaces the association by an instrumental act. The figure also shows that the internalized process remains mediated.

The subject internalizes modes of operations and objects by developing so called higher psychological functions. Every living being has natural psychological functions that coordinate its processes (e.g., nerves inform about the condition of parts of the body). The adaptation of such natural psychological functions to interact with a cultural artifact like a tool has been coined the development of higher mental function [292].

The development of higher mental functions not only enables the application of certain mediators but influences an individual's interaction with the world as a whole. The work of Luria provides a good example for the emergence of higher mental functions for written language and logico-mathematical operations [167]. Luria showed that the internalization of language and logico-mathematical operations changed the perception of the environment and the categorization processes applied to the environment.

The creation of higher mental functions depend on the higher mental functions already developed by the subject (e.g., a child is not able to learn reading without having learned the concepts of spoken language or—in case of deaf children—of sign language). AT applies a spatial metaphor to describe the development capabilities: the zone of proximal development. There may be a distance between the subject's capabilities and the required mental function to execute an activity with a mediator. The zone of proximal development specifies an area in which support bridges the distance, possibly realized by education as external guidance. Over the time, the individual step by step understands the function and finally internalizes it.

2.1.2 Activity Hierarchies

Activities do not emerge autonomously but are generated. Sources of activity generation are needs and other activities. Therefore, activities are generally included in hierarchical relations of activities. AT specifies different layers within these hierarchies to distinguish between different types of abstraction (e.g., the difference between chopping a tree and the movement of the saw) and different types of a subject's mental and physical involvement within the activity (e.g., the first time one tries to drive bike and the well trained, nearly automatic driving of a bike). In the following, needs as root cause of activities and the different layers of activities are explained in detail.

2.1.2.1 Activity Hierarchy Root: Needs

The introduced concepts of AT do not provide an explanation for the origin of activities. This is addressed by needs as root causes of activities. Objects are objectified needs which makes needs the ultimate cause behind human activity: "Any activity of an organism is directed at a certain object; an 'objectless' activity is impossible" [153]. Two types of needs can be identified, biological and psychological needs. Biological needs are the requirements of organisms. Psychological needs are cognitive commitments towards desired states. Following AT, the commitment strives to fill a vacuum between a given state and a desired state based on an activity.

A need is addressed by an object which is called motive within AT. A subject performs an activity to realize the object. The identification of a motive to address a need is "a moment of extraordinary importance" [152]. As long as a need is not addressed by a motive, the subject has a feeling of discomfort and searches for objectification. Once a motive is given, the object stimulates and guides an activity.

2.1.2.2 Activity Hierarchy Layers: Activity, Action and Operation Hierarchy

The previous passages have introduced activities and described needs as root causes for the objects of activities: A need results in an object addressed by an activity. Thus, needs generate objects and respective activities as units of subject-object interactions.

The activities that result directly from needs, generally deny a direct motive attainment. Even simple activities like prepare food, directly addressing a physical need generally require a complex subset of preparation activities. Even apparently simple motives disintegrate into a set of sub-objectives that are required for motive realization. The disintegration of activities based on sub-objectives results in a hierarchical relation of objects. AT distinguishes different levels within the hierarchy: Activities are addressed by actions and operations.

The three hierarchy levels of activity, action and operation hold well defined relations. Activities with motives have subordinate objects addressed as actions with goals and operations with conditions. Actions and operations are specializations of activities. They are special in two respects.

- **First,** actions and operations stand for real interactions of an individual with the world. Thus, the transformation process from a need to an activity to real interaction with the world is described within the hierarchy.
- **Second,** actions and operations include information about the way, the interaction is coordinated.
  - *Conscious coordination:* An action is a conscious coordination as the subject takes conscious decisions about the interaction process.
  - *Automated coordination:* In contrast, the operation is automated, not requiring any conscious decision making of the subject. One can say that the operation is internalized or embodied within the individual. An example is drinking: taking a glass filled with liquid, opening the mouth and swallowing the liquid generally does not require cognitive effort.

Two sources of operations exist. First, operations can result from improvisations. An improvisation is a spontaneous adjustment of an action without conscious thinking, e.g., reactions in emergency situations. Second, operations emerge as an automation of a conscious action realized by repeated execution. This learning of an action as operation may fall back to an action state.

Another transformation also exists: actions may themselves become activities, if they show a motive character on their own. On the other hand, a motive may become subordinate to another motive, thus becoming an action.

Actions and operations for similar goals may exist in parallel and are selected based on the context and individual preferences. Notably, that implies that actions do not suddenly become operations when they are decomposed, but rather that they are equivalent solution procedures [140].

The different transformation types show a vital relationship between the different concepts. Concepts on higher levels may collapse into concepts on lower levels whenever learning or routinization takes place or may also expand to concepts on higher levels [149, 111]. The subset of work activity systems of an individual follows the same structure. Complex work objectives are decomposed into smaller ones, until interaction occurs. Still, specific work processes exist which force the subject to follow specific processes. In this case, the subject loses the ability to decide on the decomposition by himself and might lose the ability to relate interaction to higher level goals or motives (this is investigated closer in section 2.3).

2.1.3 Activity Systems

Activity systems help to model real activities and are the foundation for activity analysis techniques. As the creation and analysis of activity systems plays an important role within this dissertation (see chapters 4 and 5), two examples for activity systems are provided in the following. The activity system by Leontiev integrates the basic concepts of activity, mediation and internalization and is a simple foundation to model real activities. The activity system provided by Engeström shows the extension of AT to new domains—collaborative activities in this case—and introduces activity analysis.

2.1.3.1 Leontiev's Activity System

Leontiev's activity system model shows the relation between subject, object and mediator as a triangle (see Figure 2.2).

A direct connection between subject and object refers to the transformation processes both face within the activity. The actual mode of interaction is given with two additional relations between subject and mediator as well as between object and mediator. Mediators which may be internalized as higher mental functions or external as physical manifestation mediate the activity. Thus the activity system represents the concepts of activity, mediation and internalization.

Instances of the activity system can be created by annotating the concepts with the respective entities (e.g., subject = information worker, tool = computer, object = information).

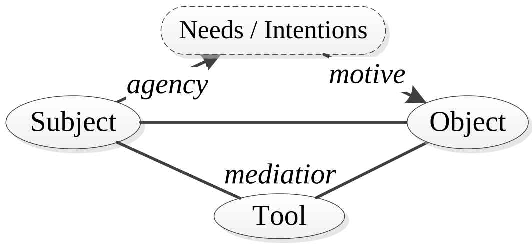

Figure 2.2.: Leontiev's activity system.

2.1.3.2 Engeström's Activity System

Engeström uses activity theory to describe and analyze activities executed within groups and organizations [84]. Therefore, Engeström designs an activity system which extends Leontiev's activity system. Next to subject and object, he introduces the community entity. For each tuple of entities, Engeström identifies a mediator. Thus, Engeström's activity system extends the concept of mediation lined out by Leontiev. In fact, rules and division of labor can, still be understood as specific types of technical and psychological tools that involve a community. Subject and community are mediated by rules. Community and object are mediated by division of labor. Finally, subject and object are mediated by tools. These three triangles which follow Leontiev's concept have been connected based on shared elements (see Figure 2.3). The mediation realized in the system is: production, distribution and exchange. Production is already given with Leontiev's system. Distribution exists between the community and an object. Exchange refers to the interaction of the subject with the community following rules.

Engeström uses the model to analyze activities and to identify tensions and contradictions within the systems. A tension or a contradiction is a negative influence one element of a system has on another element in the system. The idea of analyzing an activity system will be applied later in this thesis (see chapter 4).

2.1.4 Intermediate Result

The description of work as activity in the terminology of AT is a first step to understand work execution processes. AT provides no mechanistic perspective on work execution but describes work execution as a complex connection between a subject and the subject's environment. This perspective has been chosen due to the nature of information work. Information work is no mechanistic input output relation between subject and environment with strictly defined processes. In contrast, the development of the subject and subject's goals requires a model that captures the dynamic relationship between subject and object during work execution.

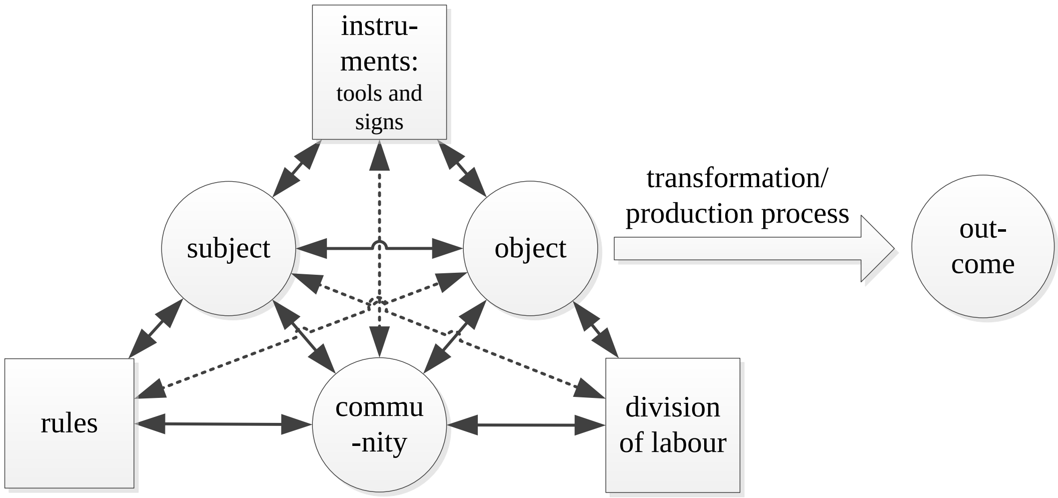

Figure 2.3.: Engeström's activity system.

To provide a basic understanding of AT the basic concepts of activity, mediation and internalization have been explained (see section 2.1.1). The concepts model the relationship between a subject and an object under consideration of the role of tools and of human development within activity execution. Different types of activities are distinguished based on the object and the cognitive involvement of the subject (see section 2.1.2). The hierarchy provides explanation for the root cause of activities in needs or in other activities. The concepts of AT are systematized within activity systems. Activity systems not only illustrate the relationships between the concepts but they also are used to analyze activities (see section 2.1.3).

Critics of AT considered the perspective to be unidirectional instead of dialectic and to be adevelopmental thus denying an explanation of emerging phenomena. The critics elaborate on an underestimated role of signs, the mind and the individual which decreases the usefulness of AT [275, 274]. A good discussion of such critics and their invalidation based on a close examination of the classical AT literature and an overview of modern AT based theory development is given by Engeström [85]. The remainder of this thesis will also show that AT shows strength when it comes to the role of the individual and the mind in work activities relevant for information work.

AT will prove useful within this dissertation as it enforces a perspective on work as dynamic processes of mediated subject object interactions which depend on the subject's personal development. The concept of activity systems is developed further and is used to analyze information work and the involved mental processes.

A limitation of the concepts provided in this section is the focus on a single activity. The coordination of different activities is not covered. This limitation is addressed within the next section which introduces action regulation theory. Action regulation theory extends AT with respect to activity organization to describe the coordination of different goals.

2.2 Psychology II: Work as Activity in Action Regulation Theory

This section introduces ART as a second contribution to the psychological perspective on work conveyed in this chapter. ART is based on AT (see section 2.1) and extends the gained understanding of an activity based purposeful and goal directed interaction with the world with respect to the cognitive processes involved in activity organization. Activity organization refers to the coordination of multiple activities and to the planning, observation and—if required—adaptation of the execution of activities. The organizational aspect of work execution is relevant in the context of this dissertation due to the relevance of autonomy for information work.

ART specifies a set of hierarchically organized cognitive processes involved in activity execution. The processes are complementary to the activity hierarchies introduced earlier (cf. section 2.1.2). The set comprises the following processes: strategic decisions, goal decomposition, operation planning and sensimotor regulation. A complex relation between the processes exists and realizes the organization of activity execution. To understand those relations, an understanding of cognitive units related to memory, perception and action is required. To facilitate the understanding, an integrated model of the organization processes, their relations and the involved cognitive units is introduced in the following (see section 2.2.1). The model is named work execution model. Based on the relations between the processes in the work execution model, the actual goal realization is described. While certain goals may coexist as abstract plans, their final realization requires that the physical activities applied to realize them get a temporal order. This becomes evident in heterachical models which are described in section 2.2.2. Having the structure of activity execution based on cognitive processes in mind, further concepts of ART relevant for activity organization are introduced:

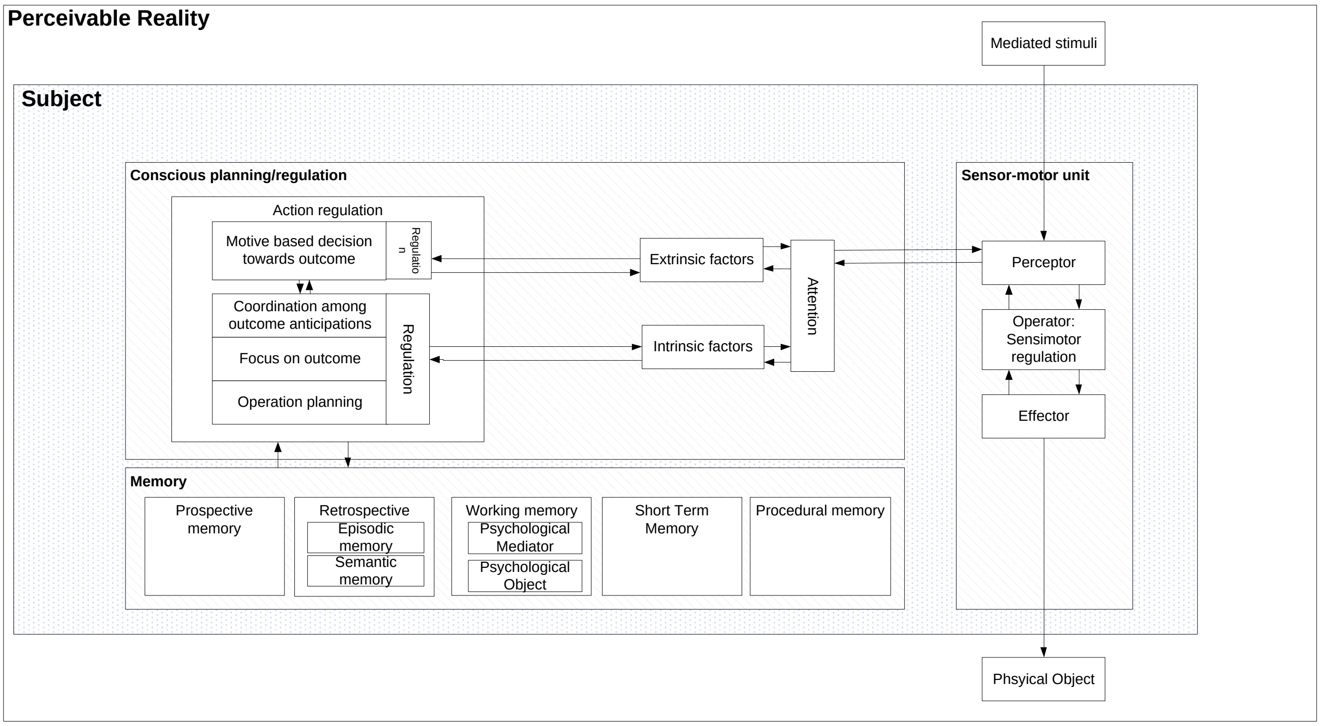

Figure 2.4.: Work execution model. A subject is embedded in a perceivable reality. Based on his sensimotor unit, stimuli are perceived and interaction with physical objects are triggered. The action regulation unit takes care of the different cognitive processes involved in goal identification and commitment in close interaction with the different types of memory. Perceived facts are filtered based on the activity as intrinsic or extrinsic factors.

- **Mental effort:** Different goals obviously are differently complex (e.g., opening a locked door with a key is simpler than opening a locked door without having the required key). ART introduces the concept of the operational cognitive image to convey goal complexity. The operational cognitive image and its relation to goal complexity is provided in section 2.2.3.

- **Regulation:** Goal realization may face difficulties and obstacles which require an adaptation of the work process. The perception of such obstacles and the propagation of the related information among the organizational processes is considered as regulative activity within ART (see section 2.2.4).

The section ends with a short introduction of another theory of work organization which is called threaded cognition. Threaded cognition offers an approach to activity execution which resembles the organization of threads within a computer system. Threaded cognition extends the gained understanding on work organization for goals of a low complexity (see section 2.2.5).

2.2.1 Work Execution Model

The work in this section discusses concepts of Hacker [108], Volpert [291] and Österreich [202] who contributed to ART. A core element of ART is the description of related organizational processes that are involved in goal identification, decisions which goal to execute and the acts involved in pursuing a goal. To understand the processes, an understanding of cognitive entities like memory and perception is required. To provide the required information to understand the processes, the work execution model is introduced. The work execution model describes goal-directed interaction between a subject and a perceived fragment of the real world and specifies the connections between different cognitive entities within the mentioned organizational processes.

The work execution model constructs a subject and the subject's relation to a perceived reality. The perceivable reality and the subject are the two main blocks of the model (see Figure 2.4). The relations action and perception connect the subject with the perceivable reality. Subject and reality are defined as follows:

- **Subject:** A subject is a self-determined entity with a physical manifestation. The subject perceives external factors and can perform actions. Figure 2.4 shows that the subject model consists of the following cognitive units: a planning/regulation unit, a perception/action unit and a memory unit. For the process of work execution, the units realize the following broad requirements:

- *Planning and regulation unit:* The planning/regulation unit takes care of the identification of goals (*Vornahme*)[1], the decision to pursue a goal (*Entschluss*) and the regulation of the execution process.
- *Perceive and act unit:* The perception/action unit encloses two processes directed towards the environment. The subject perceives the reality based on the senses and the subject interacts with the environment based on physical acts. The perception/action unit includes a direct connection between the perceptor and the effector based on the operator. This direct connection enables direct reactions to changing conditions without complex cognitive activities. Such direct reactions are automatic coordinations on operational level (e.g., breaking to avoid a collision in the car or stepping to the right to avoid collision when walking through a crowd). The connection of the sensor-motor unit to the conscious planning/regulation connects perception and action to more complex cognitive processes described later in this thesis.
- *Memory and knowledge:* A rough distinction between memory and knowledge is made. Whereas memory is more related to cognitive processes of remembering things, knowledge is a rational capacity directed towards action [146]. This understanding follows the idea of knowledge as rational capacity: a potential that manifests in action. This denies the idea of knowledge as a commodity but stresses knowledge as an intangible, individual asset of a subject that manifests in action. Polyanyi [212] distinguishes implicit and explicit knowledge. Implicit knowledge is a purely individual capacity. Explicit knowledge is abstract knowledge, models and schemes that shape the way the individual perceives the world. The complete process of work execution involves many different types of memory and knowledge. In the following, the respective memory and knowledge units relevant for work execution are provided:
  - PROSPECTIVE MEMORY: Memory of intentions, things one wants to do. Prospective memory can be triggered based on time and location. Studies have shown that location is a better prospective memory trigger than time [251].
  - SHORT TERM MEMORY: Short term memory is active and directly available. Short term memory holds a small amount of information (approx. 7 elements, $+/-$ 2) [187] in an active state for few seconds (approx. 7 seconds) [232].
  - WORKING MEMORY: The working memory contains information for a short amount of time. It allows the manipulation of this information based on transformation processes. The manipulation realizes a subject's capabilities of reasoning and comprehension [15]. The working memory is coordinated by the short term memory which organizes the information it contains and the goal directedness of the manipulation processes.
  - RETROSPECTIVE MEMORY: The retrospective memory comprises episodic and semantic memory. Episodic memory deals with remembering experiences and related facts. It holds times, places, etc. Semantic memory is the abstract knowledge of meanings, understandings and concepts which is not related to any specific experience.
  - PROCEDURAL MEMORY/KNOWLEDGE: Procedural memory and knowledge contains internalized behavior which is automatically triggered in certain situations. Therefore, it is relevant for the operator of the sensimotor-unit and not relevant for the conscious planning. In terms of AT this memory holds operations while actions are generated based on an interplay of the sensimotor unit with the conscious planning. The modification of procedural memory based on conscious cognitive processes is very complex (e.g., changing the way one holds a cup will require much attention as one will frequently fall back into the earlier habit).

  The presented memory types are concepts which provide explanation on how different cognitive processes work. The actual coordination of the brain differs and the clearly distinguished memory types presented are an effect of the coordinated interaction of different brain regions.

- **Perceivable Reality:** The existence of an objectively existing reality is assumed. A subject is part of this reality and is able to interact with it. Interaction includes perception and action. The perceived reality is the locus of action and perception. A subject is not able to perceive the complete reality. Based on the senses of sight, hearing, touch, smell, and taste a fragment of reality is perceived including, e.g., colors, sounds, textures, etc. Subjects use memory to extend the perception based on experience. Action as well as perception may be partly or completely mediated by a single tool or a set of tools as described by AT.

These basic elements are used in the following to describe the realization of goals.

2.2.2 Goal Realization Heterarchy

Subjects identify goals based on motives to satisfy their needs (cf. section 2.1). The process of goal realization is a complex individual interaction with the world. ART elaborates on the realization of gaols based on a heterarchical model[2] that decomposes goals into

[1] The German terms used by Hacker are given in italics to clarify the meant concept.
[2] The term heterarchical is used to note that subordinate structures in the hierarchy may have effect on ordinate structures.

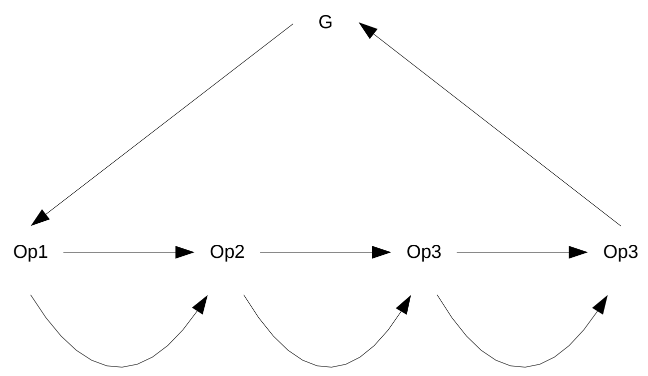

Figure 2.5.: Cyclic units of goal realization (G=Goal, Op=Operation). Taken from [108].

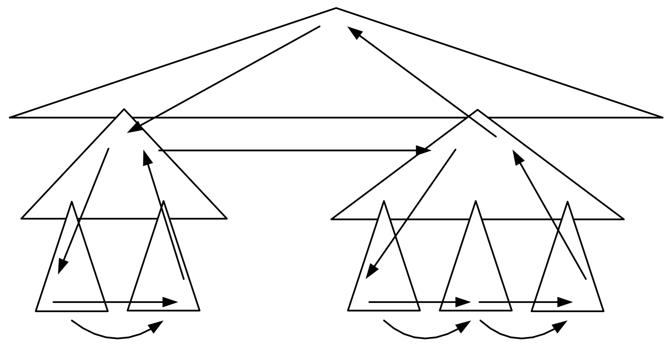

Figure 2.6.: Heterarchical structure of goal realization. Taken from [108].

subgoals. A goal anticipates a transformation of reality which requires action of the individual towards this reality. An idealistic goal realization can be described as cyclic unit [291] (see Figure 2.5). First, a goal is mentally decomposed into required operations to realize it (straight arrows in Figure 2.5). Then, the individual interacts with the world and realizes the goal (round arrows in Figure 2.5). Finally the subject checks if the result of the interactions actually fits the goal.

To model the processes of complex goal identification, decomposition down to the actual execution, heterarchical-sequential models are used. The structure has been introduced by Hacker [108] (see Figure 2.6). All nodes, with the exception of the leaf represent cognitive goals that are connected in the sense of a hierarchical goal decomposition. The leaves temporalize the hierarchy, as actual executions have a real occurrence. The goal anticipations include characteristics for errors, adaptation demand and execution. These elements make the heterarchical structure dynamic, as the heterarchy is not completely known a priori but contains placeholders that are filled while progress is made for the temporalized execution process.

2.2.3 Goal Complexity and the Operational Cognitive Image

A simplified perspective on activity execution assumes that a goal realization is directly feasible; interactions can be planned and executed. More complex goals might resist an immediate realization but require different subgoals. Only parts of the process might be anticipated because the outcome of subgoals might be unsure or a concrete planning does not make any sense (e.g., if the execution is far in the future).

Different concurrent goals that might belong together in goal, subgoal relations are coordinated based on anticipation. The operation cognitive image[3] [108] is a goal anticipation. An operational cognitive image connects perception, goal and memory in the process of execution planning by minimizing the required effort. Only those elements of the anticipated goal that are relevant for the very next process step are an element of the operational cognitive image (for more details on the operational cognitive image and its coding, see [108, p. 195]).

The cognitive process that coordinates goal realization based on an operational cognitive image can be considered as a feedback-cycle between the anticipated goal, action and perception. To transform a fragment of reality to the anticipated goal, the individual

[3] operatives Abbild

performs operations and monitors the progress and effects. The term monitoring stands for the perceptor receiving sensor information from the environment. Attention connects the perception with the goal, i.e., that attention is organized based on goals and separates the perceived reality into extrinsic and intrinsic factors that are enriched based on knowledge. Extrinsic factors are those elements of perception that are not related to a goal. Intrinsic are those elements that are related to a goal.

2.2.4 Goal Realization Regulation

This passage described the organization of goal execution between the different described cognitive processes in detail. The structural formation of a goal, operational cognitive image with feedback towards attention is a regulative-functional-unit.[4] Such units exist on different levels of the work heterarchy and expose work execution as a regulative process.

The heterarchy has different levels, from mere cognitive levels to actual interactions with the world. To describe the heterarchy of goal-subgoal relations that emerges in work execution, different taxonomies for functional-regulative units have been proposed (see [202, 291, 108]). The considered taxonomies have in common that they are closely related to Leontiev's [154] activity, action and operation hierarchy mentioned earlier (see section 2.1.2.2). Here, the five level concept proposed by Volpert is used [291] which is integrated in the work execution model.

In the following, the cognitive processes involved in goal execution are further described. The process describes the coordination of different goals with the focus goal as the goal the subject has decided to actively work on. The description explains the complex decomposition of mental goals (see Figure 2.4):

- **Sensimotor regulation:** Sensimotor regulation is the lowest level of regulative functional units. Operation programs are executed that are deeply internalized by the individual and require only a few conscious regulations. Regulation is a subconscious process that is directly connected to the perceptor. Therefore, this element is excluded from the action regulation unit in the model but is integrated in the sensor-motor-unit as an operator. As an effect, regulative processes can be triggered very quickly to directly react to changed conditions on operational level (e.g., if an object quickly approaches the subject, the subject jumps away and does not reflect the situation).

  Example: Typing text on a keyboard. While typing without watching the keyboard, a feeling of discomfort sometimes informs that the wrong key was typed. As an effect, the backspace key may be pressed to delete the typo without thinking about it. Similarly, while writing this text, the key combination Ctrl+S may be hit after each sentence without thinking about its meaning anymore.

- **Operation Planning:** Different planned operations are coordinated and sent to the sensimotor level. The temporal order of these operations is of high relevance to realize execution plans that are composed of interconnected requirements.

  Example: Hitting keys to compose words from signs that represent a text which needs to be written. Aspects like the program used to write the text, the style, the sentences, the words and the proof reading are combined in a weak planning. Weak planning means that all elements are known and are connected but no strict plan of the point in time to execute an element exists. The process emerges based on the activation of sensimotor operations. If an operation fails, a sentence cannot be finalized as information is missing, conscious replanning is required.

- **Focus/Cognitive goal decomposition:** A subject might have a set of goals he concurrently follows. Decomposition into subgoals is realized for each goal. The subject then decides which goal will be actively pursued as a focus goal. This decision—which goal becomes the pursued focus goal—is often addressed as Rubikon-decision [114]. In the planning process, alternative plans are consciously known. Planning means that a rough understanding of transformation processes required to realize the goal exists. The transformation processes have requirements towards work techniques and the reality that need to be met.

  Once the actual decision for a realization plan is taken, all alternative plans are out-of-scope. Only if the operational planning shows defects of the chosen plan, the alternative plans become conscious again. The decisions may involve complementary operational elements of different concurrent plans, i.e., that certain goal realizations share different means or requirements and, therefore, plans can be generated that address more than one goal concurrently. Time plays a coordinative role in the decision towards a plan to realize the focus goals. However, time is only an abstract concept on this cognitive level used to coordinate the availability of time as a resource and to coordinate sequence planning along temporal dependencies (e.g., opening a door before going though it).

  Example: The focus goal is to write an article about the activities within a project. Writing the article has become the focus goal, as its priority is higher than the priority of other elements. Alternative goals are a telephone call to a colleague, writing an email to the manager and to drink a cup of coffee. A deadline is the reason for the high priority of the text document. During the process, the goal of drinking a cup of coffee is activated, as well. A cup of coffee stands next to the workplace

[4] For a detailed description of regulative-functional-units, their different types VVR and ZMB and their relation to TOTE (test-operate-exit) units, see [108].

and the subject sips from time to time. Adaptations are possible; if the telephone rings, goal coordination will be triggered to decide if answering the phone has enough priority to react to it. An exception is when answering the phone is automated. Taking a phone call might be an automated sensimotor action which does not involve conscious thinking.

- **Goal coordination:** Goals that are kept in prospective memory are triggered by temporal or situational events. Once a goal is recognized, the individual re-evaluates the active focus goals and needs to decide if an interruption of the active goals is required. Each time such a decision process occurs, certain goals need to become part of the prospective memory with new triggers. Although strict requirements towards techniques and reality conditions are considered in the planning on the focus goal level, less granular aspects of techniques and reality requirements are considered during the goals coordination. For the goal coordination, attention is regulated and resources are provided to those concurrent goals that are followed on the focus goal level. Temporal order is a relevant fact without being very specific.

  Example: An Outlook pop-up signifies that a project report needs to be sent to the manager within the next two hours. This triggers the coordination among different goals. Focus goals must be interrupted to meet the timeline.

- **Strategic decisions:** Strategic decisions form the highest level of regulative-functional units. The subject identifies goals or declares his commitment for delegated goals. The temporal order is of minor importance.

  Example: Some time ago, the motive of deciding what to do after finishing formal education leaded to a job at a project management office. The job is connected with a contract and the contract specifies different objectives that need to be reached to be successful. The subject identifies goals and is committed to projects based on those objectives. The contract leads to a strategic direction coordinating the subject's perception and acceptance of goals.

For the hierarchy of goal coordination, different aspects need to be considered. First, within a goal hierarchy, each mentioned goal coordination level may result in more than just one layer: A goal on goal coordination level may be decomposed into subgoals that, still belong to the goal coordination level.

Second, the relevance of time increases the closer a level is to the bottom level. On the highest levels of the hierarchy, time is merely a vague concept for resource planning and to order dependencies among different goals. With each step, time becomes more important. On the lowest level, timing is so relevant that the regulation process is excluded from conscious processes. Regulation on the sensimotor level is affective and, therefore, directly connected to perception.

Third, the different regulative unit levels help to understand the complex interplay of goals and subgoals. The feedback relation between different operational cognitive images becomes obvious: if a change on any level occurs, it is likely that the regulation is propagated to the enclosing goals on the same goal type level or on different goal type levels of the heterarchy.

2.2.5 Related Concept: Threaded Cognition

Threaded cognition is a model-based integrated theory of concurrent multitasking [231]. Concurrent multitasking addresses the problem of subjects being involved in multiple goals at a time. Consequently, the organization between the different goals needs to be structured. Threaded cognition analyzes organization mechanisms, involved resources and their availability and the resolution of resource conflicts. Based on the Act-R model [7], computational models can be generated to provide explicit predictions about multitasking, the involved processes and time consumptions.

2.2.5.1 Goal Realization

The execution of multiple goals is represented as the execution of multiple parallel threads that are coordinated by a processor and use different processing resources. Tensions emerge from parallel threads requiring the same resource at the same time or from resources being required by more than one thread.

Salvucci and Taatgen give a good overview of the basic assumptions of threaded cognition in [231] which is summarized here. The resources in threaded cognition are cognitive, perceptual and motor resources. Conscious coordination is realized by the cognitive resource that includes separate procedural and declarative resources. Declarative resources stand for declarative memory, information chunks that can be recalled. The procedural resource stands for skill and capabilities. Declarative and procedural resources make the distinction between knowledge and memory. New information is acquired based on required action in the environment.

The core assumption is that subjects commit to a set of active goals. Each active goal has a thread that works on the available resources. Resources can only be accessed by one thread at a time. Therefore, tensions based on resource requirements occur. Multiple requests for the same resource are solved by resource alignment to the least recent thread.

2.2.5.2 Interaction of Goals

These elements show that the model does not assume a hierarchy of goals on different processing levels. The procedural resource coordinates between the different threads. Complex goals are decomposed into a number of threads. Still, the complexity of decision making with goals that have effects on other goals seems to be difficult to model using threaded cognition. In contrast, threaded cognition is very comprehensible when it comes to goals that do not have complex connectivity, e.g., routine goals. Operationalization (see section 2.1.2.2) is also well modeled in threaded cognition.

2.2.5.3 Critical Assessment of Threaded Cognition

ART as well as threaded cognition help to understand the individual reasoning about motives and the creation of actual interaction with the work based on motives.

Threaded cognition analyzes multitasking and occurring effects. One important difference between ART and threaded cognition is the goal structure. ART assumes a hierarchy of goals based on the AT hierarchy of activities, actions and operations. Threaded cognition assumes a flat structure composed of a set of parallel threads.

It has become apparent that both approaches have advantages when it comes to explaining different cognitive processes. The application of ART enables explaining complex cognitive work regulations, e.g., the meaning of a failure is tracked back to its meaning in the context of an activity and different operations are executed in parallel with the purpose of reducing execution time. Threaded cognition helps to understand interruption processes and goal switches from the perspective of time consumption and optimization.

2.2.6 Intermediate Result

This section specified activity organization based on ART. Activity organization includes the coordination of multiple activities and the planning, execution and regulation of activities. These aspects address a limitation of AT. AT focuses on singular activities and the mediated relation between a subject and an object. ART focuses on the cognitive processes of the individual which are relevant for the commitment to a work process and for the execution of the work process. The work execution model has been introduced to describe the cognitive processes in relation to cognitive entities and the real world (see section 2.2.1). The actual process to pursue a goal is a heterarchical structure which applies the cognitive processes (see section 2.2.2). ART rejects a mechanic perspective on work execution. Therefore, the heterarchical structure is coordinated based on a cognitive plan, the operational cognitive image. Cognitive dissonance between the operational cognitive image and actual perception triggers regulation (see section 2.2.4 and 2.2.3). The concept of regulation deserves specific attention. Regulation introduces a dynamic relation between the elements of the heterarchy. In this respect regulation is a concept to consider conscious decisions and observed conditions within work execution processes.

The presentation of ART concludes the psychology perspective on work. Overall, a holistic perspective on work execution by a subject has been provided. An understanding of work fundamental for this dissertation has been provided. Work is composed of activities that mediate between subject and object. Activity execution is hierarchically structured by cognitive processes. The processes take care of the decisions for goals, the planning, the coordination and—if required—the adaptation of a work execution process.

2.3 Organization Theory: Work as Contract based Commitment Organized by a Division of Labor

The previous sections have provided a psychological perspective on work as activity. The identification of activities, the individual commitment and the organization of activity execution has been described based on AT and ART. In the following, work as activity is set into the context of wage labor and division of labor. Work is no longer individual but considered as delegated and controlled to a higher or lesser extent within a community. The goal is to extend the gained understanding of work with respect to modern working conditions in a market economy. This understanding provides important insights into the effects of wage labor and division of labor on the work execution by a subject. The presented perspective is taken from organization theory. Based on a contract based commitment to an organization, the individual becomes part of complex, value-creating processes.

To understand the effects of wage labor and division of labor on the individuals' work execution, there is a precondition to be considered. The design of work processes in divisions of labor with respect to the formalization degree and the freedom of action given to the worker needs to be specified. Therefore, a design spectrum for work execution ranging from a high degree of autonomy to rigid heteronomy is provided (see section 2.3.1). Based on this design spectrum, two aspects of the work processes of the individual are important for the course of this dissertation. First, the effects of different designs of the work process on the individual. Dependent on the degree of autonomy and the complexity of work, the cognitive processes involved in work execution highly differ. This is described with recourse to ART and in reference of the design spectrum (see section 2.3.2). Second, the design of work processes generates the specialization of work. The subject has a specific job to do with specific characteristics and constraints. There is a tendency for naive categorizations of specializations, e.g., distinctions between symbolic work and manual work (for a discussion, see [145]). Such a perspective is disadvantageous with respect to the goal of this dissertation. Here, the design of the work process is a primary criteria for the specialization. As a result, a work spectrum with a focus on autonomy and heteronomy results (see section 2.3.3).

2.3.1 Work Design: Between Autonomy and Heteronomy

When it comes to the division of labor, the provenance of motives and the reason for the goal commitment is not obvious. AT states that motives are derived from needs still holds. Nevertheless, the actual provenance of a goal within a division of labor is not obvious. Leontiev shows the complexity of goal provenance when he describes the activity of hunting in a group: the activity of each group member seems senseless, unless the activities of all group members are taken into account [153].

Wage labor means that subjects make contract-based commitments to organizations. The contract generally assigns a functional unit and a role to the subject. Based on the contract, the subject becomes part of a hierarchy of authority and responsibility. One reason for the subject to sign the contract is the optimization of a need for necessities. In exchange to the support of the subject, the functional unit and the role transfer goals to the individual.

Hacker distinguishes two types of goals in the context of division of labor [108]. Adopted goals are those goals that are transferred to a subject. In contrast, self-identified goals are identified by the subject. Hacker states that self-identified goals have a higher comfort for the individual. For adopted goals an individual has to decide to adopt the goal. Only if the goal is adopted, the goal can be considered like a self-identified goal. The knowledge about the goal heavily influences the chance that adopted goals are well perceived. For value oriented work, the goals are identified by the individual most probably resulting in a stronger goal commitment of the individual.

The way goals are transferred to the subjects highly differs. Picot distinguishes four degrees of goal communication which are closely related to the autonomy given to the subject [210, p.234]:

- **Processes:** A detailed description of a work process is given to the subject. The subject has to realize the process and has no abilities to modify any element in the process.
- **Procedural guidance:** Goals are communicated to the subject and subjects have to realize the goals by following procedural guidance.
- **Outcome description:** Based on an outcome description the individuals need to decide how to attain a goal.
- **Value orientation:** Subjects identify goals compliant with the organizational values and objectives, and control the goal realization on their own.

Hybrid forms of the different formalizations most probably exist. The given types obviously have an enormous impact on the way the subject achieves goals.

2.3.2 Work Cognition: The Influence of Work Design and Complexity

The previous section has shown that the subject involved in the division of labor has different degrees of freedom with respect to the design of the work process. While goal delegation based on value transfer provides freedom of action, a goal delegation based on process transfer strictly specifies what the subject has to do when and how.

The cognitive processes involved in the process of pursuing a goal depend on the freedom of action and the resulting performance complexity. If the process is highly specified and the performance required to pursue a goal is relatively simple, the subject follows the work process with little cognitive involvement. If the process is highly specified but the required performance is complex, the subject builds an idea of the process result and reviews the progress based on an operational cognitive image.

A high freedom of action results in decision processes. The subject needs to decide how to pursue the goal. Next to the decision process, operational cognitive images of different complexities are required, based on the required performance quality. The resulting dependencies between cognitive processes, performance complexity and freedom of action are provided in table 2.1.

In the next chapters, the relevance of this for this dissertation will become obvious. The characteristics of information work result in a high performance complexity and a high degree of freedom. Therefore, information work generally means that complex cognitive processes are required. A subject involved in information work needs to plan the work process, review it closely and be prepared to react on an unforeseen situation.

2.3.3 Work Spectrum: A Classification based on Work Design and Complexity

The division of labor in wage labor results in a specialization of the work activities subjects have to execute. To analyze work, it is necessary to be able to compare different types of work based on relevant criteria. A frequently used criteria is based on the subject matter of work [145], separating symbolic work from material work. The previous sections do not provide any hints for such a perspective. In contrast, two criteria have emerged which seem to be much more applicable. On the one hand, the process design: The degree of autonomy the individual has during work execution. On the other hand, the performance complexity. Both criteria have been described as influencing factors for the overall goal complexity and as being different means of cognitive activation of the worker (see section 2.3.2).

Performance requirement complexity \ Degree of Freedom	**without (completely algorithmic)**	**with**
Complexity low: no path or result anticipation	Delegated and applied quantitative goals: experience temporal progress to the goal	Independent quantitative goals: conscious decision towards goal realization and of the realization process
Complexity high: with path and/or result anticipation	Delegated and applied content goals: experience progress of content towards goal	Independent goal identification and goal anticipation with respect to: 1. sequence 2. path, means 3. goal characteristics, consciousness of goal and problems

Table 2.1.: Goal complexity, table taken from [108].

These criteria are used in the following to classify work. To show the usefulness of a classification on the two criteria, different examples of work that occur in shared labor are presented. The examples show a spectrum of work and have been chosen to show the benefits of the chosen classification criteria and to show additional reasons to reject a distinction of work based on the subject matter:

- **Software Engineer:** The software engineer has an extensive formal education. He applies the gained knowledge in the conception and realization of software artifacts. For this, he applies trained work patterns and mixes them with experience and creativity to create solutions. The main working instruments are various computer tools and their representation of information.

  The software engineer mainly works with signs and has to solve different programming tasks. The work process is mediated by outcome descriptions, requiring extensive individual planning by the individual. Still, the software engineer needs to follow certain predefined processes (e.g., application of programming patterns or usage of a specific infrastructure). Thus, decision making is of high relevance, outcome specification dominates. A high performance complexity is given due to the large amount of complex symbolic tools. Still, different elements of the work are composed of processes and process orientation with a low performance complexity.

- **Administrative clerk:** The administrative clerk has some formal education. His task is the application of guidelines and predefined processes to cases. A case is an abstract representation of an occurrence or status in the real world. The freedom of action as well as the decision space is limited. Nonetheless, creativity and experience is required to realize how guidelines and processes cover actual cases. The main working instruments are paper-based and digital forms.

  The administrative clerk is organized based on processes while value orientation is relevant, too: The creation of cases requires abstraction and problem solving if the occurrence or status in the real world is not directly addressed by an existing process. Overall, the performance complexity is relatively low as the tool set is limited and completely predefined. The dominance of process work shows that working with signs does not necessarily mean knowledge-intensive work but may also follow strict processes without much freedom.

  An interesting aspect is that the administrative clerk, although mainly using psychological tools, has very limited freedom. This is an additional argument against a separation between manual and cognitive work.

- **Brick layer:** The brick layer realizes architectural plans under the supervision of a foreman. His education is gained during an apprenticeship. Based on the structural circumstances he has to adjust the requirements in discussion with the foreman.

  The brick layer works process oriented, sometimes also outcome oriented. The performance complexity is relatively high due to the complex combination of different tools in the complex setting of a building lot. The work process is specified to some extent. Nevertheless, he uses individual experience and skills to optimize the working process, identify problems and adapt his work accordingly.

- **Machine operator:** In the following a machine operator is considered, trained to observe a machine while it performs a certain task without additional freedom. The performance complexity is low due to the limited set of tools and interaction processes. Based on different indicators the operator decides if the machine operates well. Little formal education is required.

  The machine operator is strictly guided by processes. He has nearly no freedom during the execution of his work. He is an example of an extreme position of process orientation.

The examples show that each type of work is a combination of different types of complexity and different degrees of autonomy. Extreme cases of raw process organization and low complexity (classic assembly line) and raw value orientation and high complexity (entrepreneur) exist. Nevertheless, it is assumed that a large amount of the workforce is subject to hybrid formalization that mixes value orientation, outcome specification, process orientation and processes. Nevertheless, the classification provides a basic scheme to analyze the ratio of autonomy and complexity and get an understanding of the considered work. Additionally, the examples show that a distinction of work based on the subject matter should be rejected. There is no connection visible between the cognitive effort and the subject matter.

2.3.4 Intermediate Result

The organization theory perspective enables the analysis of work executed in a division of labor as it is given in a modern market economy.

In recourse to ART, the effect of different work process designs for the cognition of the subject has been described (see sections 2.3.1 and 2.3.2). This connection between work process design and the subject's cognition shows that the process design needs to be reflected in each analysis of work. Within this dissertation, it will be used to analyze information work. To enable this analysis, a classification spectrum of work has been described (see section 2.3.3). The classification focuses on the two mentioned criteria: the degree of autonomy and the complexity. The subject matter of work (e.g., work on symbols or manual work) is not relevant for the classification.

The next section will show how work process designs based on heteronomy—namely bureaucratic work process design—has been a key success factor within the industrialization. A limitation of autonomy has benefits with respect to planning and quality reliability, and is an important success factor in shared labor. The effect is a weak individual commitment towards the delegated goals.

2.4 Sociology: Work as Means of Coordination and Control in the Information Society

This section provides a sociologic perspective on work. This perspective provides an explanation of specific types of work based on conditions and requirements of societal formations. Thus, work is identified as a product of society which requires an understanding of the environment which demands a work type. This section is based on Beniger's "The Control Revolution" [25] and Castell's "The Information Age: Economy, Society and Culture" [47], acclaimed work on societal transformations over the past 300 years and the fostered work types. Both provide explanation for the emergence of an information society which generates information work. Thus, the last section of this chapter finally introduces information work based on an explanation of the societal conditions which generated this type of work.

To investigate into the emergence of specific work types based on societal circumstances, two processes are of utmost importance: coordination and control. Only based on coordination and control productive interplay of a society can be achieved.[5] To introduce coordination and control, both are defined first (see section 2.4.1). The next step is to show how a control crisis is addressed by work types which present themselves as techniques to reestablish coordination and control. This is shown by the example of different control crisis which emerged with the industrial revolution. Rationalization was an effective solution to different control problems which generated a bureaucracy with a heteronomic workforce (see section 2.4.2). The emergence of the information work workforce follows a similar logic (see section 2.4.3). The globalization resulted in a crisis of coordination and control which required a workforce which was able to react quickly to changing conditions. A workforce with a high degree of autonomy and new means of information access, production and exchange was capable of solving the crisis, the information work workforce. In this sense, this section introduces information work as a work type to control and coordinate global processes of commodity production and exchange in an unstable environment.

2.4.1 Coordination and Control

Coordination and control are fundamental requirements for productive processes. Coordination is "the organization of the different elements of a complex body or activity so as to enable them to work together efficiently" [205]. The purposeful organization of elements is a basic requirement for productive processes. To realize coordination, control is required. Here, control is understood as purposive influence on things towards a predetermined goal [25, p. 7]. In this sense, control is a general concept, enclosing absolute control as well as weak, probabilistic control (e.g., purposive influence on behavior).

Shared labor distributes logically connected work tasks among different subjects to realize a goal. Coordination and control are required to integrate the logically connected work tasks to realize the initial goal. The higher the complexity of the organizational structure, the higher the demand for coordination and control [113].

[5] The terms coordination and control are used broadly, also including schemes like Smith's invisible hand [263].

Information is crucial for the realization of goal directed action, based on coordination and control. The etymology of control as being derived from *contrarotulare*, to compare something "against the rolls", against the records/duplicate accounts in ancient times, shows this connection. In this sense, control maintenance is inseparable from a societies' communication techniques.

2.4.2 Bureaucracy and Rationalization as Control Techniques

An increased complexity of the system creates the requirement for an increase of coordination and control. Rationalization and bureaucracy are basic techniques to realize this.

Rationalization decreases the amount of information to be processed in a system. Bureaucracy generalizes *the specific* to an impersonal case structure. These techniques have been extensively applied to address a crisis of control that emerged when technical innovation changed the processes in the fields of production, distribution and consumption of goods and services. Technological innovation extends the processes that sustain life but it also increases the complexity of the processes as well and generates an increased demand for control.

Beniger [25] describes the use of railroads and other steam-powered transportation in the 1840s as begin of a control crisis. Transportation increased the speed of transportation and laid the foundation for the industrial revolution and mass production. At the same time, the existing infrastructure was not capable of coordinating the increased speed, requiring new innovations. A cycle of innovation and control resulted. Innovations improved the transportation infrastructure and generated new types of control crisis. Distribution needed to be coordinated and controlled. Mass production required market integration and control of demand and consumption.

Next to technical innovations, rationalization and bureaucracy were useful control techniques to address the different crises. Organizations were structured hierarchically and the tayloristic perspective created the idea of an exchangeable workforce as resource. Different productive industries created structures to tackle control problems (e.g., the the assembly line). Transportation networks were organized by systems of signals and signs. Commercials aimed to gain market control. These few examples stand for the continued resolution of control problems by technical innovations, often in the spirit of rationalization and bureaucracy and going together with an increasing relevance of information (for a detailed description, see [25]).

The increasing relevance of information to realize coordination and control created the information society. The first to identify and analyze the transformation of the workforce on a large scale was Fritz Machlup [168]. Machlup showed that 25 % of the GNP of the US was in a sector he considered representative for what he called the knowledge society (including media, education etc.). This increase of information was closely connected with rationalization and bureaucracy. Bureaucracy originally relied on paper-based information systems based on records and files. Typewriters, stenography and other techniques improved the creation of information. Communication technology like telegraph and later fax increased the speed of information distribution. Nevertheless, it was the use of the computer that fundamentally changed the techniques of information creation, storage and its consumption. Computers like the LEO1, the Univac or the IBM 1401 were the machines that made computing techniques available to companies [164].

For large parts of the workforce, a limited influence on the organizations can be stated as they did not make decisions that had influence on the organization. The logic of the organization was in the hands of few and materialized in the physical setup of companies and the logic internalized in the workforce. The structure followed tayloristic and bureaucratic principles. Processes were strictly defined in hierarchies that were created by few entrepreneurs.

2.4.3 Limits of Rationalization and Bureaucracy

The means of production reached a limit in the 1970s. The result was a commercial crisis. Castells [47] argues that the improvement of production and the increase of the sales volume demanded a fundamental change of the production and sales structure of the existing companies. Company structures changed to less vertical integration and more supply chain orientation [47]. The implementation of different changes was the beginning of a societal transformation on a global scale [25, p. 6]. Information exchange and flexibility became increasingly important and an economy organized in networks of commodity and information exchange on a global scale [25] emerged. Globalization happened, which can be seen in the increasing amount of border crossing transactions in the second half of the twentieth century [116].

Benigers' discussion of the control crisis gives a good explanation of the role of information and communication technology in the crisis [25]. The globalization responded to a crisis and generated an increasing demand for control that required the global distribution of information in real time. Consequently information and communication technology was of high relevance which possibly influenced the design of computers. Computers are rational and bureaucratic machines. The digital coding logic of the computer is an impressive example of rationalization. The formalization requirement to describe entities and their interconnections as abstractions in data structures and algorithms follows a bureaucratic logic. In this sense, the computer as semiotic machine of sign transformation responded to the increasing amount of information generated with control and coordination mechanisms. And even more, the later development of the computer to a communication device which operated on a global scale responded to requirements which emerged from the control crisis. Castells [47] localizes the integration of information and communication technologies in the productive sector mainly in the 1970s in the context of the financial crisis. In the 1980s the new technology was slowly adapted and in the 1990s it was productively used within the organizations. After the productive phase, the innovation in the sector addressed newly emerging control problems, e.g., by introducing the integrated business information system in the 1990s.

The communication technology is only one aspect of the transformation. In fact, the societal transformation required a new workforce to address the global distribution of financial, productive and distributive processes based on real time information. Coordination and control based on information requires a trained workforce which processes the information, maintains the exchange networks and immediately acts regulative once conditions change—the information worker. Bureaucratic structures that control individuals based on strictly defined processes are less adaptive and responsive to changes.

Surprisingly, a logic of increased rationalization generates a very different workforce, a workforce which requires the individual to take over responsibility. Consequently, Drucker demands a workforce which is capable of identifying the required tasks autonomously and of controlling their own productivity: "It demands that we impose the responsibility for their productivity on the individual knowledge workers themselves. Knowledge Workers have to manage themselves. They have to have autonomy"[6] [78]. Information work is a type of work which is largely composed of knowledge work and heavily relies on interaction with models of real world processes and object status data, presented in signs, symbols and speech acts, accessed, modified and distributed with information technology [145, 213]. Information work is a modern control technique which is embedded in organizational structure but requires a high degree of autonomy, resulting in controlled autonomy.

2.4.4 Intermediate Result

The work executed within a society is not anachronistic but results from the society's requirements and conditions. Thus, work is element and product of a social period. This conclusion is the main result of this section. The result is applied to the type of work this thesis deals with, information work. The information worker is revealed as being a product of a society with globally distributed productive processes.

This result is concluded based on the work on social transformations by Beniger and Castell. The need to establish coordination and control within a society generates specific types of work. Coordination and control have been introduced (see section 2.4.1) and the generation of specific work types based on social conditions has been discussed with respect to the industrial revolution (see section 2.4.2). The industrial revolution triggered a crisis of control which was addressed by rationalization. The rationalization mainly relied on bureaucratic and thus inherently heteronomic types of work. Rationalization failed to address the requirements of real time decision making in a globalized, real time economy (see section 2.4.3). Information work addressed the requirements to some extent based on autonomy. Goals are organized to a large—although not exclusive—extent by outcome description and value orientation. Information work applies rationalization but mainly on the level of data processing embedded in the logic of the computer.

The increased importance of the individual can be seen as an important opportunity. The relevance of the individual grows. The individual becomes an organizational asset and has the freedom of planning the personal work processes.

2.5 Summary

This chapter has provided the foundation for the further treatment of information work and respective memory threats within this thesis. Only based on a broad understanding of work, the specific characteristics of information work become apparent. The complexity and the broadness of the concept of work in general and the connections of information work to many different domains are the reasons for the presentation of work based on different scientific perspectives.

The main contributions of the provided perspectives for this thesis are summarized in the following:

- **Psychology (see section 2.1 and section 2.2):** The psychology perspective is a foundation to understand work as a productive activity. Activities are a complex concept. Activities not only involve the physical interaction with the world. They also involve the cognitive processes which take care of the planning and the regulation of the physical interaction and its perception. Here, AT and ART are used to provide a holistic perspective on activities which includes cognitive and physical processes within activity execution, as well. The main contributions for this thesis are models to explain activity processing, namely activity systems and the work execution model. Both guide the analysis of activities and are required for an analysis of information work (see chapter 3). Additionally, AT and ART are foundation for a system design method created within this dissertation (see section 4).

- **Organization theory (see section 2.4):** The organization theory perspective analyzes common aspects of organizations and their optimization with respect to criteria like efficiency and productivity [304]. Here, organization theory bridges the gap

[6] Drucker refers to the emerging workforce as knowledge workers, a term frequently used. Here, it is assumed that knowledge work is just a dimension of work (for a detailed discussion, see [145], also cf. spectrum of work 2.3.3). Knowledge work is considered to be the representation of decision making in work. An activity which is and has always been relevant for many different types of work. In this respect, knowledge work is an anachronistic concept and can be found in very many different types of work. Information work emerged in the context of the information society at the end of the 20th century and addresses the specific requirement of fast and responsive coordination and control.

between the subject with personal goals and the organization which consumes the subject's productive capabilities. The subject becomes an element of the organization and commits to specific activities following principles of shared labor and specialization. Thus, the activities of the subject are delegated to a certain extent by the organization the subject works for. The degree of specialization and the degree of autonomy or heteronomy the subject faces is of specific interest for this dissertation. In this respect, the classification of work based on performance complexity and freedom of action is the primary contribution of the organization theory perspective.

- **Sociology (see section 2.4):** The sociological perspective provides reasons for the emergence of information work based on social requirements. The perspective especially addresses the challenge of not considering work types anachronistically but as being the product of social systems which exist at a specific period of time (e.g., assembly line work is the product of the industrial revolution as a societal transformation process). Based on the analytical and empirical work on social transformations by Beniger and Castell the systemic relations between work and the society are identified. Work is described as a productive element of social systems which addresses requirements of a society as a whole. In this sense, information work emerged to address requirements of a globalized society. Situating information work in a social system with access to specific technologies and with a global logic of market economy is the contribution of the sociological perspective for this dissertation.

To conclude, the discussion of work provides a rich overview of different aspects of work relevant for this dissertation. Explanations for different aspects of work are provided. This covers explanations for the work execution of an individual (psychology), explanations for the work organization in a division of labor (organizational perspective) and the reasons for the emergence of specific types of work. Work becomes a central connection between a subject, productive organizations and society as a whole. In this sense, work generates the individual in productive processes which stabilize the social reality. Based on this logic information work emerged from the social reality of the late twentieth century to address requirements, for example the need for responsiveness to changing conditions within globally distributed productive processes.

The next chapter analyzes information work based on the background of the gained understanding of work.

3 Information Work Ideal Type

In the previous section different perspectives on work have been presented to gain a generic understanding of the concept of work. This chapter builds on the gained understanding and elaborates on information work. In order to identify the relevant aspects of information work in general and for memory threats in information work in particular, the scope of this broadly used term will be limited to an ideal type[1].

Many investigations into information work exist [78, 69]. A selection has been synthesized to an ideal type by Pyöriä [213]. Pyöriä's ideal type comprises the following characteristics: 1) Education: information work requires formal education and on the job training 2) Skills: information work requires the transfer of skills 3) Nature of work: information work has a low level of standardization 4) Organization: different organization types are applied ranging from bureaucracies to self-managing teams 5) Medium of work: information work is applied to symbols and other people. However, Pyöriä's ideal type has a bias towards being a "common denominator" of information work theories from an organization theory perspective. This dissertation requires an ideal type with a bias towards the subject's cognitive involvement in information work to reason about memory threats. Therefore, an ideal type with a focus on cognitive processes involved in information work execution is provided in the following. The ideal type especially focuses the computer workplace.

The first contribution to the ideal type is an outline of basic characteristics of information work (see section 3.1). Next to obvious aspects like the relevance of information and the application of information and communication technology, the outline shows a striking characteristic. Information work generally offers a high degree of autonomy with respect to the design of the work process. Thus, autonomy affects information work coordination. Therefore, the second contribution to the information work ideal type is an analysis of information work coordination (see section 3.2). The analysis will show that information work is coordinated based on interruptions. Interruptions may be triggered by the subject or may be external interruptions, the subject has to react on. The challenge of the presentation of interruption based work coordination is to capture the ambivalence of interruptions. While interruptions are the major and thus important mean of coordination resulting from the high degree of autonomy, interruptions at the same time threat information work execution as they are a source of memory threats.

To prepare further investigations into the role of interruptions for memory threats, the third contribution to the ideal type is a framework to model information work execution. The idea is to identify basic units of work the work processes are composed of (e.g., reading and writing are generic actions which are adjusted to fit specific requirements of an activity). A literature review shows that the idea of such units of work is quite common. Nevertheless, existing taxonomies of basic actions for information work processes are not directly applicable within this dissertation due to different limitations (e.g., no focus on the computer workplace, no property to connect cognitive processes and actual interactions). Those limitations are addressed in this dissertation. An empirical study is reported which identifies a taxonomy of basic actions, information work is composed of, referred to as knowledge actions and desktop operations (see section 3.3).

The ideal type developed in this chapter goes beyond the commonplace understanding of the topic. The core element is a model of the execution process based on interruption based coordination and the application of recurring basic actions. This model especially unfolds the complex circumstances which result in memory failures in information work.

3.1 Ideal Type: Basic Characteristics

The first contribution to the information work ideal type is the introduction of basic characteristics. As the ideal type stresses the relevance of the cognitive processes which orchestrate the activities the terminology of AT and ART is suitable and used in the following. Information work presents itself as a mixture of value orientation and process specification (cf. section 2.3). Therefore, the goal of the ideal type is to show the influence of value orientation and the resulting autonomy on information work execution. The subject of the activity faces autonomy by designing the activities based on personal knowledge and perceived circumstances. To reason about the resulting work processes, effectiveness and efficiency are suitable concepts. Effectiveness is the capability of producing an intended outcome within an activity. Due to autonomy the subject selects goals and specifies acceptance criteria autonomously, thus influencing the effectiveness (see section 3.1.1). Efficiency as ratio between input and output refers to the personal development of the subject. Autonomy requires the subject to coordinate and control the work process and optimize the required effort in relation to the output (see section 3.1.2).

Another relevant element of the ideal type is the object of the activity. While the relevance of information within in information work is no surprise, the actual role of information is more complex. Information is the object of work which is transformed into a

[1] An ideal type is "formed by the one-sided accentuation of one or more points of view and by the synthesis of a great many diffuse, discrete, more or less present and occasionally absent concrete individual phenomena, which are arranged according to those onesidedly emphasized viewpoints into a unified analytical construct [...]." "[A] utopia [that] cannot be found empirically anywhere in reality" [296]

work result during an activity. For information work at the computer workplace, the encoding of information based on symbols and their persistence in information objects is of specific relevance (see section 3.1.3). A threat for the work execution is the amount of information objects involved in activities. The mediator of information work activity is the computer workplace. The following presentation illustrates the relevance of the workplace based on empirical data (see section 3.1.4).

3.1.1 Effectiveness

Effectiveness is the capability of producing an intended outcome within an activity.

To specify effectiveness in the context of information work, it is important to understand the way an outcome is produced. Information work is a mixture of autonomy and heteronomy. Some goals pursued are delegated based on processes and process guidance; others are only structured based on outcome descriptions or value orientation. Strictly formalized work processes are exceptions. The low formalization degree enables quicker reaction to changed conditions and improves the individual performance as individual optimization is not constrained by predefined processes. As an effect information work involves the subject in an ongoing individual negotiation which goals to commit to and how to realize them. Although the subject might be aware of numerous goals, limited capabilities and resources restrain the amount of goals that can be realized. Therefore, the individual selects a subset of the identified goals based on priorities and appropriateness and commits himself to their realization. The prospective work on achieving a goal based on an activity is considered as a task which helps to classify and structure upcoming activities [24, 62, 99, 95]. Tasks are executed by activities. The activities transform an object into an outcome. Effectiveness is only given if the outcome of the activity is a sufficient realization of the goal (see Figure 3.1).

Information work frequently is wage labor and depends on delegated goals. Nevertheless, the delegated goals frequently have a high degree of abstraction (e.g., create an algorithm while considering a set of constraints OR increase sales in a specific region). Thus the subject needs to create feasible subgoals and needs to specify acceptance criteria for those subgoals by himself. Therefore, the constraints of success for many activities executed by the subject are specified by the subject. For sure, in the long run the overall success with respect to the delegated goals will verify the individual acceptance criteria. Nevertheless, on a daily basis the subject individually specifies personal acceptance criteria, is able to modify personal acceptance criteria and, therefore, autonomously specifies the personal effectiveness.

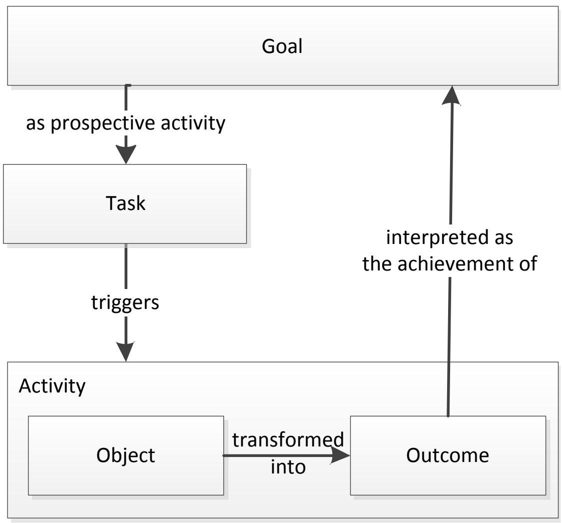

Figure 3.1.: Idealized relations between goal, task, activity, object and outcome. An outcome does not necessarily meet a goal, resulting in complex regulative activities.

As long as the subject is convinced that an activity can be finalized successfully, the respective activity exists. If a transformation of the object into the outcome becomes unlikely, the individual can assure the successful execution based on regulations. Regulations present themselves as shifting and deprecating activities in the activity heterarchy. Only if the individual is convinced that it has become impossible to successfully finalize an activity, the activity fails. Therefore, effectiveness is closely related to the individual assessment of anticipation fulfilment.

The process is as follows: As soon as the goal criteria or conditions on any level of the heterarchy are perceived as being endangered, the regulation process is initiated. Assuming that an activity on any level of the heterarchy is threatened the regulation triggers a modification of the operational cognitive image. The modification might have two results. First, a simple result: a modification on the threat level or below is successful as fewer or additional subgoals are chosen or the anticipated goal on the level of the threat is modified. Second, a complex result: the threat is propagated to the parent level if the modification on the threat level and below fails.

If the threat is propagated to the parent level, a regulation is triggered again to transform the operational cognitive image which, in case of failure, propagates the threat to the parent goal. The process is continued until the individual identifies a heterarchy level that allows a regulation of the operational cognitive image with a confidence of goal realization. As a consequence, all subgoals are deprecated or rededicated.[2] This failure propagation demonstrates the adaptability of the *planning/regulation* unit which realizes failure as search for an alternative. Problems of this heterarchy address the willingness of individuals to drop goals. The higher the manipulations in the heterarchy are the higher the expected resistance, as more and more goals are threatened by deprecation.

The search task is a good example for failure as a decision. The subject identifies an information requirement. Tools are identified that may provide information, e.g., search engines. The subject starts with an initial set of keywords. The subject reviews search results and tries to improve the keywords. Based on titles and short descriptions resources for a closer review are selected. Based on the success of the search method the individual step by step improves the understanding of the personal knowledge demand, decides on time-constraints and time distribution among query, review and read activities. If an ongoing search fails to provide results, adaptations of keywords and the search process are made until the subject decides that he will not identify the required information (For a detailed investigation of the search task, see [43, 42, 41]). If the required information is not accessed, the goal heterarchy is used as follows. The problem is propagated to the level above the search in the heterarchy. The subject needs to ask himself “do I really need this information and if yes, are there other ways to get it?”. If there is no way of accessing the required information, the next parent level is threatened and the reason for the information requirement needs to be regulated, again. The process continues until an activity is successfully regulated or the whole activity is deprecated.

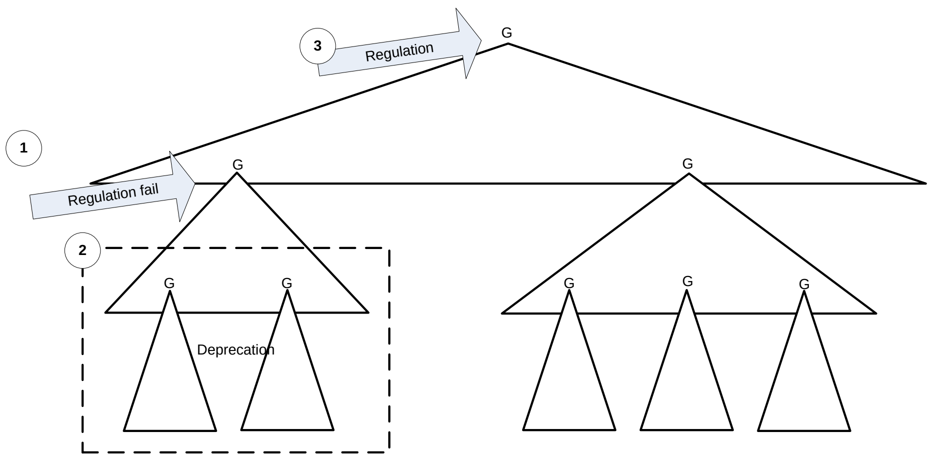

Figure 3.2.: A perception triggers a regulation on G22. The regulation is identified as a failure, as criteria or conditions of the goal have become unattainable, even if the work is adapted. Consequently, the subtree of G22 is deprecated and G11 requires adaptation.

3.1.2 Efficiency

Efficiency is the ratio between input and output. For the subject the inputs are time and resources. The subject invests work on tools like the computer for an amount of time to create a result. ART supports an understanding of time and resource investment as a result of more complex cognitive processes of commitment and planning. From this point of view, efficiency is closely related to the ability of the subject to commit to motive related goals by orchestrating and regulating the work done. Efficiency addresses the quality of planning and regulating processes. The central instance for planning and regulation are operational cognitive images which structure the work (see section 2.2).

A high efficiency thus requires a good operational cognitive image which covers a deep knowledge of the anticipated outcome, ways to fulfil the anticipation and adaptations to address unexpected events during the work execution. Additionally, an awareness of the priority of each activity is necessary, combined with an awareness of all existing activities. Work needs to be coordination between the different activities to meet deadlines or to profit from situational opportunities. Thus, efficiency is a combination of coordinative overview and the operational cognitive image/knowledge of each activity.

[2] An individual might fail to identify all subgoals, especially, if subgoals have become habits. The effect would be the execution of work which has lost its initial reason.

3.1.3 Relevance of Information

Information work outcome as well as raw material is information [9]. First, information work produces information as an instrument for illocutionary and perlocutionary acts in Austin's sense [11]. The individual executes an act by creating a certain piece of information (illocution)[3] or the individual disseminates information (which can also be the modification of symbols in computers) to have a following effect in the real world (perlocution). Second, the environment the information work is executed in is complex and subject to frequent change and transformation processes. To rule out uncertainty, the subject involved in information work maintains mental models of the world and of the effects of interaction with the world. The mental models support the work process. Therefore, the consumption of information to maintain and shape the models is a crucial element of information work. The consumption as well as the creation of information can be referred to, using the terms internalization and externalization (cf. [212]). The maintenance of mental models is the modification of higher mental functions based on the internalization of information (see section 2.1). The subject involved in information work is a consumer and a publisher in personal union who shapes his experience in the processes of information interaction.

Despite the relevance of information, its extent quickly results in negative effects for the work execution often referred to as information overload. Information overload addresses two different problems as a study by [86] showed. On the one hand, the volume of information (stated by 79 % of the study participants). On the other hand, the difficulty or impossibility of managing the information (reported by 62 % of the study participants). A study between 1997 and 1999 at Fortune 100 companies gives an idea of the amount of information and the rapid pace at which the amount increases [94]. In 1997, the average employee sent and received 178 messages and documents a day, in 1998 the volume increased to 190 messages and documents and in 1999 the amount was 201 (in Canada, it was only 156 and 169 in 1998 and 1999). People who execute information work mainly used telephone calls, email, voice mail, interoffice mail/faxes and postal mail with an average of nine different communication tools. The documents themselves are not always newly created, but they have a complex provenance, as copy, paste and other types of information reuse result in an enormous information exchange between ostensibly different mails and documents [137].

3.1.4 Relevance of the Computer Workplace

Information cannot be considered without the tools or mediators that make it accessible. Within information work, information and communication technology is the most important mediator of information which supports processes of internalization and externalization (cf. [212]) and the related maintenance and use of higher mental functions like mental models. Information technology enables real time distribution of large amounts of partially automatically produced information (e.g., sensor information) which supports the quick responsiveness required in the information society [47] (cf. section 2.4). Yet, information technology is a mediator. As Bødker terms it: "people are not interacting with computers: they interact with the world through computers" [28].

Information technology used in information work presents itself as a device mix. Several studies on information work exist. Gonzalez and Mark analyze information work with respect to tools that are used and with respect to the structure of the workdays [101, 174, 175]. Bellotti examines the way people organize themselves [24, 23], to name only a few influential researchers in the domain. The existing work on information work not only proves the relevance of the information and communication technology by describing frequent switching between physical and digital artifacts [101, 132]. Studies also show that 50 % of the working day is spent with information and communication technology. For the used information and communication technology, the computer workplace has outstanding relevance as it is used 37 % of working day.[4]

Typically, computer workplaces are multi-purpose workplaces, providing a set of software applications with numerous functionalities to enable and support a large variety of information creation and consumption activities. Individuals blend the use of software applications in individual processes of work execution which manifest expertise as well as experience.

3.1.5 Intermediate Results

The presented characteristics are the foundation of the ideal type developed in this chapter. An activity-centric perspective on information work has been used to show the influence of value orientation and resulting autonomy on the subject.

The activity perspective follows the basic subject, object mediator relation. Interestingly, the freedom to identify goals and design actions has relevant influence on the success and quality of activity execution in terms of effectiveness and efficiency. Effectiveness and efficiency directly depend on the subject's perception of the activity and the respective capabilities to trigger regulation once the activity execution is threatened (see section 3.1.1 and 3.1.2). The object of the activity is information encoded in information objects. Empirical studies do not only show the relevance of information but also show that large amounts of information threat the successful execution of information work activities (see section 3.1.3). The computer workplace mediates the interaction with information (see section 3.1.4). The mediator provides capabilities of accessing, modifying and disseminating information. Empirical studies show the

[3] An example is a priest who contracts a marriage.

[4] Study with 13 participants: 6 business analysts, 4 developers and 4 managers. The relevance of smartphones in nowadays work structure suggests that the whole device usage has increased, but that the PC and the smartphone together are the two most important devices.

relevance of the computer workplace for information work in general. While the autonomy of information work has been highlighted, the computer rationalizes the execution as the interaction follows strictly predefined rules and requires specific workflows to achieve specific goals.

These characteristics support the understanding of information work activity and the cognitive processes involved in information work execution. Action regulation theory (see section 2.2) identifies cognitive processes involved in work execution based on goal complexity and work guidance (see table 2.1). The complexity of the mediator (information workplace), the object (information overload) and their dependence on effectiveness and efficiency of information work have been presented in this section. Thus, a high goal complexity for information work follows. According to the complexity specification (see table 2.1) the following cognitive process applies: independent goal identification and goal anticipation with respect to: sequence path, means, goal characteristics, consciousness of goal and problems.

The following sections show how the information worker deals with the resulting complexity by elaborating on the cognitive processes involved in information work execution. The next section describes the cognitive processes involved in the coordination of information work execution.

3.2 Ideal Type: Information Work Coordination

The information work ideal type developed within this dissertation is intended as analytical framework for information work execution. The basic characteristics provided in the previous section specify information work in an activity-centric terminology and underline the cognitive complexity of information work execution. This complexity is investigated further by considering the coordination of different activities in the following. Activity coordination will give first explanations for memory failures.

Subjects involved in information work do not process goals sequentially. To execute information work, people multitask which means that many different tasks are executed in parallel or in rapid succession [159]. Multitasking is largely coordinated by interruptions. Even before a goal is completed, an interrupted may occur which forces the subject to decide about the parallel activation of another focus goal or to switch the focus goal. Interruptions in the context of human-computer interaction have been extensively studied. For an overview, see [40] and for an introduction, see [180, 179, 181].

What are interruptions? Definitions of interruptions roughly fall in two categories. First, interruptions can be considered an events that happen to an individual. Second, interruptions can be considered as processes that involve the interrupted individual. The dictionary defines interruption as “to stop or hinder by breaking in”, “to break the uniformity or continuity of”,“to break in upon an action” [183]. This dictionary description presents interruptions as events subjects must passively admit to. This passive admittance is a perspective that is used in interruption research that focuses on interruption as an externally triggered event, specifying it as a “synchronous interaction which was not initiated by the subject” [201]. In contrast, a process perspective on interruption shows the active participation of the subject in an interruption, defining it as “process of coordinating abrupt changes in people’s activities” [179]. Brixey extends the process perspective with respect to situatedness of interruptions and interruption source: “Interruption [...] [is] a break in the performance of a human activity initiated by a source internal or external to the recipient with occurrence situated within the context of a setting or location. This break results in the suspension of an initial task to perform an unplanned task with the assumption that the initial task will be resumed” [35, 36]. In this dissertation, the process perspective is followed as it stresses the individual ability to shape an interruption. As it is assumed that more than one focus goal can be active, it is additionally assumed that an interruption might result in a parallel activation next to activity switch or switch refusal.

Arguments exist which consider multitasking and interruption as inherently negative [227] because they threaten focused work. Despite the negative perspective, the following section shows the relevance interruptions have for information work execution. First, the relevance of interruptions for information work is specified (see section 3.2.1), then the process involved in interruptions is presented (see section 3.2.2) and the different types of interruptions are specified (see section 3.2.3). Interestingly enough, interruptions are not inherently negative but also have positive effects on work execution. A literature review of psychological studies shows this. Nevertheless, first evidence for the role of interruptions with respect to memory failures is provided as well (see section 3.2.4). Finally, means of changing the relevance of interruptions or supporting the interruption process are discussed (see section 3.2.5). This section will show that interruptions are a necessary requirement of information work and will argue for mechanisms to support the cognitive processes which are triggered based on interruptions.

3.2.1 Interruptions in Information Work

Interruptions occur frequently in information work. A study among Fortune 100 companies [94] showed in 1999 that 84 % of the surveyed staffers are interrupted at least three times per hour by messages. In this group, 51 % are interrupted six or more times per hour. Seventy-one percent feel overwhelmed by the message traffic. Nevertheless, interruptions are not inherently negative, as they are used to coordinate goal realization.

Czerwinski [65] reports on an average of 50 goal shifts over the week that were relevant to realize complex goals. Most shifts were triggered by interruptions. Apart from coordinative interruptions, interruptions may as well provide necessary information that is required to realize a goal [190, 101]. In this sense, interruptions may even be a core characteristic of work, as Sproull identifies for managers [268].

Interruptions at the computer workplace have become increasingly relevant with the computer becoming a multitask machine [232]. Single task computers discouraged multitasking, whereas the ability to start multiple programs at the same time and to access multiple possibly permanently updated information sets at a time encouraged the work on different focus goals in parallel or in rapid succession. Thus, the computer provides different types of information that may distract the user and may trigger interruptions, e.g., instant messaging [66]. Remembering computer work is especially complex, as remembering goals (prospective memory) for important computing events is fragile [67] and forgetting intentions in demanding situations is rapid [82].

3.2.2 Interruption Process

An interruption is a complex cognitive process that can be activated arbitrarily often for any uncompleted focus goal. Each interruption is triggered by an event, a prospective memory trigger or an external trigger. For the individual the interruption presents itself as information which does not belong to the focus goal, therefore, requiring an activity switch. An activity switch is a decision process involving tasks competing with one another like stimuli do [261]. The focus goal addressed by an interruption needs to be recalled to decide whether it is ignored, activated in parallel to the existing work (if possible) or if the work switched.

If a switch is performed, the initial focus goal is rehearsed. Rehearsal is the process of memorizing the status of the focus goal realization (i.e., relevant aspects of the operational cognitive image are memorized). Rehearsal is important even for short interruptions, as only two seconds of information can be rehearsed in a phonological loop [14]. An item in working memory can be remembered roughly seven seconds [45]. In the case of a switch, rehearsal starts at the onset of the target goal of the interruption and continues for a short time afterwards, executing concurrently with the target goal of the interruption [189]. Indication exists that the rehearsal may take place in parallel with the activation and the start of the new focus task [232]. Even if the interruption is known beforehand, the time consumed by the rehearsal remains the same [121].

In context with the activation of an additional or a different focus goal, a transformation of the environment may be required. Thus, interruptions include on the one hand cognitive turnover times for rehearsal (in case of a switch) and time for construction/recall. On the other hand, the environmental needs to be arranged accordingly.[5] For the computer workplace, different applications may be required for the new focus goal, or different information objects need to be accessed.

An interruption which results in a focus goal switch (no parallel execution) has been described by Trafton [276] who used four events: (1) an alert for a secondary task, (2) the begin of a secondary task, (3) the end of a secondary task and (4) the first observed primary task action. Trafton focuses on actual activity switches. Based on these events, Trafton defines three timeframes: the interruption lag between (1) and (2), the interruption between (2) and (3) and the resumption lag between (3) and (4). Here it is assumed that the resumption lag can be dropped due to the facts that (1) only 40 % of all post-interruption tasks are actually continuing the initial focus task and (2) the activation of a new task after the interruption might have the characteristic of an interruption again.[6]

3.2.3 Interruption Types

Based on the provenance of the interruption trigger in an interruption process (see 3.2.2), two general types of interruptions are distinguished [232, 101]: internal and external ones. External interruptions result from events in the environment. Internal interruptions come from our own thought processes. Different studies have shown independently that interruptions are evenly distributed among internal and external interruptions (Gonzalez et al. [101] talks about 50 %, [65] talks about 40 % self-initiated interruptions).

- **Internal Interruption** An internal interruption is initiated by an individual based on prospective memory. Prospective memory triggers goal remembrance based on time and location (studies show a better recall for location [251]). This trigger enables internal decisions about commitment towards identified goals, work start for committed goals or resumption of goals interrupted earlier. In this sense, internal interruptions present themselves as side effects of internal background operations outside the attention towards focus goals [188]. The decision is taken on the *strategic decision* layer in the *planning regulation* unit of the work execution model. Such decisions integrate into the personal strive towards the optimization of goal commitments and focus goal decisions with respect to priorities and failure avoidance.

  The subject decides on his own to be interrupted. Therefore, there is a high probability that the interruption target is perceived as having a high importance at the moment of interruption or that the interruption is not harmful (e.g., phases of lower cognitive activity: the last focus activity was just accomplished).

[5] In general, the recall and the construction of the environment will depend on each other, as a recalled fragment results in access of an object which again supports the recall, and so on.

[6] Research on interruption in ACT-R theory is important in this context. ACT-R theory describes a problem state as set of information that is related to a problem. Despite the differences between action regulation theory and threaded cognition, the problem state is closely related to the operational cognitive image. Whereas the problem state is limited to some data points (e.g., numbers and operators in a calculation task), the operational cognitive image includes additional information. Still, both capture task relevant information and demand the memorization and activation of this type of information in an interruption process. The initial focus goal related operational cognitive image/problem state is memorized and the new focus goal operational cognitive image/problem state needs to be reactivated.

Different types of self-interruptions have been identified in a self-interruption typology [138]. The typology consists of seven self-interruption related categories related to computer activities. These are in particular: environment change (adjustment), information search to facilitate the focus goal (inquiry), switching to alleviate fatigue or frustration (break); recalling an unrelated goal (recollection) or perceiving a cue to another goal (trigger); performing activities out of habit (routine); or filling idle time (wait). The aforementioned prospective memory-related self-interruptions are addressed by adjustment, trigger, recollection, wait and inquiry. The remaining three tackle the cognitive state of the subject (break, recollection and routine).

- **External Interruption** An external interruption is caused by an event external to the subject that is an impediment for the focus goal realization. The external event is not related to the focus goal realization and a continuation of the focus goal realization is not possible, enforcing the subject to interrupt himself. Examples are notifications [180] or communication interruptions [66]. Many external interruption definitions do not require the interruption to be an impediment to the process to pursue the focus goal (e.g., [188]). Here, this aspect is considered crucial to avoid overlaps between the internal and external category. Seeing a colleague and being reminded of a goal, or looking at your watch and being reminded, both are internal interruptions of the type adjustment. External interruptions in contrast would only occur if the interaction is enforced, e.g., a colleague asks something and will not leave until he gets a reply.

  The external interruption has a higher complexity than the internal interruption. The interruption is triggered by an external stimulus which consumes the attention of the individual. The subject needs to recall which topic or activity the interruption addresses. There are three specific threats for external interruption :

  1. *Inappropriate situation:* An external interruption might occur at an inappropriate moment in time, i.e., the interruption has a very negative impact on the work executed at the moment of interruption.
  2. *Unknown interruption target:* It is unclear if the interruption goal is known by the subject. Therefore, the subject might not be able to address the interruption immediately.
  3. *Priority disregard:* The subject might not be able to forcefully reject the interruption due to the strength of the stimulus (cf. a colleague stands in the door and asks a question. Sending the colleague away might have negative consequences,). Therefore, the interruption might modify or ignore existing priorities of the subject.

  Each type of event can be the cause of an external interruption. McFarlane [181] lists sources for external interruptions in human-computer interaction like: another person; computer; other animate object; inanimate object.

3.2.4 Effects of Interruptions

Interruptions are considered to be an important coordinating mechanism of information work. Despite this positive perspective, interruptions have multi-faceted impact on work execution as they require recall and often rehearsal of work executions by interfering with the focus goal. Research shows the impact as being dependent on the individual and on the task [31, 139, 10, 305].

Positive effects of interruptions have been identified with respect to recall for the interruption of a simple task, addressed as the Zeigarnik effect. The Zeigarnik effect states the ability of remembering interrupted tasks better than tasks that were completed without interruption [303]. This was specified by van Bergen as requiring the tasks to be engaging and the subjects to be motivated by the instructions [278]. Performance improvements have been described for simple, non-challenging tasks. The individual's performance was lowered, as unused cognitive capacity was used to think about non-task-related things. Interruptions required participants to focus more deeply on the initial focus task which resulted in better overall human performance [266, 267]. The performance increase is not given for the interruption of a complex or cognitive-demanding initial focus goal.

Negative effects of interruptions have been identified with respect to prospective memory, mental tension, mistakes and performance. Prospective memory is especially threatened by interruption. Only 40 % of the time interrupted tasks are resumed immediately after an interruption [65] and the likelihood of successful execution of the interrupted task decreases [67, 201, 251]. Mental tension has been described twofold. On the one hand, unpredictable and uncontrollable interruptions induce mental tension (addressed as stress) [57]. On the other hand, the need to speed up work to eliminate time lost with interruptions may cause stress [175]. The probability of mistakes for a new focus task and for a resumption of the initial focus task later increases [100, 50, 65]. The consumed time increases if an individual is engaged in complex or cognitive demanding tasks while an interruption occurs [266]. Gillie shows that a post interruption task, independent of the decision taken in the interruption, is solved slower, especially if the task considered in interruption and the initial focus task share many similarities [100].

3.2.5 Interruption Timing and Process Support

Triggers for interruption processes are distractions. As seen, external events create internal as well as external interruptions, demand recall and result in an evaluation of the interruption and may result in a focus goal change. The effects have shown that few situations exist where interruptions have a positive impact on work performance. Interruptions might provide additional information that interferes with the focus goal as well as with the strategic planning. In contrast, many negative impacts of interruptions have been identified, i.e., some interruptions are more disruptive than others which is dependent on factors like the individual, the focus goal and the interrupting goal.

Evidence exists that interruptions are more disruptive while subjects experience higher mental workloads [1, 17, 66, 17]. This depends on the rehearsal, decision and activation processes involved in the interruption process that are more complex when complex operational images need to be maintained or multiple priorities need to be compared.

For interruptions, it is not a question of how to avoid them or how to judge the quality of these interruptions. Interruptions exist and are a necessary element of work execution. The question is how to help people to deal with interruptions, how to simplify the parallel activation of switching between different activities by improving recall and rehearsal of activities. Another aspect is the identifications of the right moments for interruptions. The optimal point of time for an interruption and the support of the interruption process by tools emerge as two important support fields to alleviate the effects of interruptions.

3.2.6 Intermediate Results

This section has introduced interruption based coordination (see section 3.2.1). The contribution of this section is the analysis of interruptions based on literature from psychology and operation theory. Important insights for the treatment of the topic were gained. Notably, two commonplace assumptions about interruptions need to be rejected:

- **First**, interruptions are no unnecessary burden but a necessary element of information work. Based on internal and external interruptions activity switches are realized. The switch is executed based on complex mnemonic processes. The subject memorizes the previous activity and recalls or creates information about the next activity (see section 3.2.2). Two sources of interruptions exist, the subject himself (internal interruption) or the external world (external interruption) (see section 3.2.3).

- **Second**, interruptions not necessarily have negative effects on work execution. The subject sometimes feels more engaged due to interruptions. Still, interruptions frequently trigger stress, decrease the subject's performance and most importantly for this thesis, they result in memory failures (see section 3.2.4).

The important conclusion of this section is that interruptions trigger memory failures if the subject switches between activities which are complex and which have few things in common. Nevertheless, due to their relevance, interruptions cannot be avoided. The relevant question is how to support interruptions (see section 3.2.5).

Interruption based coordination enriches the activity-centric perspective of the information work ideal type. Interruptions are a basic element of information work execution. Yet, interruptions are a source of memory failures. This conclusion motivates further research into support mechanisms for interruptions. Research which is conducted later within this dissertation (see chapter 5). The next section completes the ideal type by specifying units of work, information work processes are composed of. This is required, as interruptions are embedded in work execution processes and knowledge about those processes is required to truly understand the way interruptions work.

3.3 Ideal Type: Information Work Techniques

This section investigates into the information work process and will show that the work process is composed of recurring work techniques. The understanding of the work process gained in the previous sections suggests that activities are very diverse. This is not completely true. Although objects are different and no two activities are exactly the same, it can be assumed that certain regularities exist which apply for all activity execution processes.

The idea of the following section is that logical units of work are applied during the work process which may share similarities. Such logical units of work share similarities because they emerge from the same recurring work technique. It is assumed that work techniques—like templates or patterns—are generic to fit to a large set of work situations. Work techniques are well trained and deeply internalized by the individual. An individual tries to address goals by applying those work techniques to the problem domain generating a logical unit of work.

Information work at the computer workplace limits the domain under consideration. Standard applications and interaction metaphors provide a stable work execution environment: standard applications offer a set of functions for generic information transformation while the interaction follows a similar structure: menu bars, context menus, files and folders, etc. Therefore, the computer workplace seems to fit the use of work techniques that can be used in different situations very well.

This section investigates into information work techniques. First, a literature based review of existing taxonomies of recurring logical units of information work is provided (see section 3.3.1). The literature review supports the general idea of recurring work techniques while it also uncovers different flaws within the existing taxonomies. Due to the flaws, a reuse of the existing taxonomies within this dissertation is avoided. Therefore, the remainder of the section investigates into recurring work techniques to provide a respective taxonomy to be used within this dissertation (see section 3.3.3). The taxonomy is identified based on empirical research (see section 3.3.2).[7]

3.3.1 Recurring Information Work Activities in the Literature

In the domain of management and organization studies, different authors have described recurring types of activities within information work[8]. Researchers hope that an investigation into the sources of information work productivity within the work processes is possible based on the identification of such relevant activities [70, 78]. This section provides an overview of the research as a starting point for the work on logical units of work. The presentation shows that this type of investigation is no established research domain but emerge in very different scientific areas (e.g., computer science, organization theory). Consequently, the scattered contributions focus on different aspects of work and do not share a terminology. Terms like activities and actions are used in reference to a commonplace understanding. Despite these flaws, the following list encourages work on the identification of recurring logical units of work. All contributions identify a taxonomy of activities (the taxonomies are provided in table 3.1) and provide relevant insight into information work:

- **Knowledge management:** A large body of research analyzes information work activities with a focus on the interaction with information. Markus et al. [176] described a series of necessary activities for re-using knowledge in an organization with reference to Davenport [68] such as documenting knowledge, packaging knowledge for re-use and disseminating knowledge. Barth et al. [22] define a personal knowledge management process model that is centered on knowledge activities and also mentions tools that can be used. The activities are: accessing information and ideas (desktop search), evaluating information and ideas (collaborative filtering), organizing information and ideas (diaries, portals), analyzing information and ideas (spreadsheets, visualizations), conveying information and ideas (presentations, web sites), collaboration around information and ideas (messaging, meeting), securing information and ideas. The model of Geisler [97] is based on interviews with managers who were actively engaged in knowledge management in their organizations. He works out four stages of what he calls "knowledge processing", namely generation, transfer, implementation and absorption which are analyzed further.

- **Personal knowledge work:** Völkel [290] empirically found use cases in personal knowledge work such as learning, idea management, document creation, argumentation, and personal social network management. Völkel [290] on the other hand investigated knowledge cues and processes in personal knowledge management. His knowledge model comes with seven main knowledge processes that are extended by four additional processes in collaborative knowledge work. The knowledge processes of the Völkel model are then further investigated and split in fine-grained process steps.

- **Networking activities:** Two taxonomies of information work execution with respect to the interaction between information workers have been identified. Skyrme et al. [262] identified a set of knowledge networking activities such as self-awareness, communication, and developing networks. Efimova [81] examined knowledge sharing and network development practices of information workers involved in weblog activities. She suggested personal knowledge activities that incorporated awareness, establishing and maintaining networks, and organizing ideas.

- **Work execution:** The third group of research considers a broad range of work unit types relevant for information work execution.

  According to North [199] planning, analyzing (including searching, structuring, and reflecting), synthesizing (including combination, reconfiguration, designing), communication, documentation and learning are core value creation components in information work. Holsapple et al. [123, 122] developed an advanced knowledge flow model that contains the elements knowledge acquisition, coordination, and measurement of information work. Davis et al. [72] discussed the effects of ubiquitous computing on the productivity in information work and identified affordances that provide support for the activities authoring, review, planning, collaboration, and communication.

  Hädrich [109] identified a set of eight recurring knowledge actions as particles of information work. Each activity is an abstraction from the actual task execution process and described as: authoring, co-authoring, training, acquisition, update, feedback, expert search, and invitation. Those knowledge actions have been decomposed knowledge activities into different steps.

The identified taxonomies follow the idea that certain logical units of work exist which are applied again and again to execute information work. Some taxonomies share concepts (e.g., acquisition and authoring) which show consensus on some activities although the taxonomies originated from different research domains.

[7] The author has conducted different studies in the domain of information focusing on the work technique identification. These are reported in [247, 221, 246]. In this thesis, one study which is mentioned in [247, 221] is reported in an extended form as it fits best to the described work on an information work support tool.

[8] The authors often address the knowledge worker. To follow the term usage of the thesis, the term information work is applied based on the arguments given in chapter 2, esp. in section 2.3.

Davenport [68]	Davis [72]	Sellen and Harper [252]	Efimova [81]	Holsapple and Jones [123, 122]	Hädrich [109]	Bernstein [26]
acquisition, application, creation, dissemination, documentation, packaging	authoring, review, planning, collaboration, communication	acquiring, annotating, composing, organizing, processing	awareness, collaboration, conversations, creativity, establishing and maintaining relations, exposure lurking, making sense of information, organizing ideas	acquisition, assimilation, control, co-ordination, emission, generation, leadership, measurement, selection	acquisition, authoring, co-authoring, expert search, feedback, invitation, training, update	analyzing, applying, evaluating, organizing, presenting, retrieving, securing, sharing, storing

Table 3.1.: Taxonomies of recurring logical units of work.

The taxonomies encourage the idea of basic units of work information work processes are composed of. The basic question for this dissertation is whether one of the taxonomies can be used to enrich the ideal type by a process perspective on information work execution. The following aspects need to be considered to answer the question:

- **Granularity:** Most reported units of work stand for complex activities. The taxonomies of Völkel [290] and Hädrich [109] also specify more fine grained activities which are combined in a process like manner to execute the complex activities. The taxonomies do not specify reasons for the chosen granularity but the granularity resulted from the used approaches of empirical, analytical nature.
- **Work process:** While the taxonomies contain recurring activities of information work they do not provide an understanding of the realization of the activities.
- **Computer workplace** The taxonomies originate from different research domains. None of the taxonomies explicitly focuses the interaction with the computer to execute information work.

Due to the listed aspects none of the taxonomies is directly applicable within this dissertation. For this thesis a taxonomy of units of information work is required that 1) focuses on information work at the computer workplace 2) identifies units of work that are combined to execute information work 3) provides explanation for the actual interactions that can be observed. In the following, empirical research is conducted to create a taxonomy that addresses these requirements.

3.3.2 Information Work Technique Study

The latter review of recurring activities relevant for information work encouraged the idea of logical units of work. Nevertheless, none of the existing taxonomies of such activities is appropriate to model information work at the computer workplace based on logical units of information work. In the following, empirical research is reported to develop a taxonomy that 1) focuses on information work at the computer workplace 2) identifies units of work that are combined to execute information work 3) provides explanation for the actual interactions that can be observed.

The reported study combines two approaches of investigation. On the one hand, information work execution is observed to identify the work processes exposed in information work execution. On the other hand, the participants are asked to renarrate their work processes based on their own vocabulary. This subjective perspective helps to identify units of work with the granularity generally used by information workers to classify their work execution. Combined with the observation data, the subjects' logical units of work can be associated with the interaction data gathered during the execution process. Thus, the cognitive logical units of work can be used to create a taxonomy which is connected to interactions with the computer.

3.3.2.1 Study Methodology

Twenty employees of an international software company working in the field of IT research executed predefined, knowledge-intensive tasks. A set of seven tasks was created by a focus group of four researchers. The focus group was used to the work tasks the participants of the study execute regularly. Tasks were meant to represent goals the participants are familiar with while keeping a complexity to address it as information work tasks. Therefore, tasks requiring individual planning of execution steps including the selection

Task	Description
Task 1	Provide information on related work on individual topic
Task 2	Set up meeting to discuss conference paper review
Task 3	Decide on applicant invitation and communicate your decision
Task 4	Plan a trip and inform your colleague with all involved information
Task 5	Create presentation of a paper in a foreign language
Task 6	Find application partners and experts for a research project
Task 7	Search for information on software functionality and save for later use

Table 3.2.: Study tasks.

of involved information sources and applications were generated. The seven created tasks are provided in table 3.2. A detailed description of the data set1 generated and analyzed in the study is given in the appendix (see section C.1).

Six study participants had post-doctoral positions, eight were PhD students, four were master thesis students and two were researchers. Each participant worked 90 minutes on tasks that were randomly selected from the seven created tasks (see table 3.2 for the task list). Overall 115 task instances were executed. The task execution processes of the participants were recorded during each study, using video (computer screen capturing and video recording of user face) and an application that recorded events of user system interaction (e.g., open file, focus application, etc.). The study participants were shadowed by two study organizers who made notes during the study execution and tried to get an understanding of the execution process. Participants could ask the organizers for help with respect to the understanding of a task but not with respect the to execution procedures.

Immediately after finishing a task, the respective participant answered the question "What have you done to execute the task and why?". They were asked to use a granularity "reasonable to describe the work process to another person". A structure was provided for the answer, e.g., "I *browsed detailed job offers* WITH THE GOAL/TO *examine applicants*". (italics denote participant input). Participants could mark that subtasks were executed in a loop or multiple times. This process applied a modification of the unitization method. Unitization is a technique introduced by Newtson [196] to investigate into people's segmentation of actions or movements (e.g., decomposing a movie into segments). The segmentation of an activity into a set of events requires breakpoints. A breakpoint is the point in time that marks the end of one segment and the begin of a new segment.[9] It is difficult to apply the original unitization, asking participants to segment a video of a person executing a task, to the domain of information work execution. A segmentation based on observations by others, not done by the information worker would neglect the true intention behind activities (for a segmentation study with segmentation conducted by others for information creation and manipulation tasks, see [135]). Therefore, the participants segmented their own work execution process and labeled the segments.

The study explored the segments users identified and was intended to identify regularities of segment usage and segment classification.

3.3.2.2 Result Data Description

For the reviewed 115 task executions, the participants reported 445 different work segments. The execution durations lie between 76 and 1956 seconds. The average execution time, the max and the min time is shown in table 3.4 and visualized in a scatter chart 3.4. The execution times for tasks differ between the different participants but they are normally distributed per task, as a Shapiro-Wilk test has shown.[10]

The tasks which required frequent searches and aggregations of information and decision-making, required the most time for all users. These are especially the decision for applicant invitation (task 3), planning a trip to a conference (task 4) and the translation of a document (task 5).

The application usage per task over all users was extracted (see Figure 3.3). Although user specific aspects are not visible within the graph, the overall trend of information workers using multiple applications during work execution becomes obvious. A detailed investigation on the application usage shows that for all tasks (task 6 is an exception) all users use at least three identical applications. Task six only shares two applications as most users focused on the usage of the Internet Explorer to conduct the expert search without using additional applications. A more detailed investigation of the application usage per user shows that user specific application choices are relevant and hint to individual preferences and work styles (this has been reported in [247]).

Tasks are executed by a mixture of different applications. Switches between applications occur very frequently during task execution. The data of the study includes 3229 application switches, resulting in a mean usage duration of 25.8 seconds for an application before a switch occurs (see table 3.3).

The dependence between segmentation behavior of the participants and the task or the duration was analyzed: The number of segments identified by the participants has been compared to the execution duration and the task number. Nevertheless, no significant

[9] Zacks [302] stresses that unitization needs to be distinguished from perceptional segmentation which is an ongoing process not dependent of any intentional task.

[10] Shapiro-Wilk is suitable to decide on normal distribution for small data sets (n<30) [253].

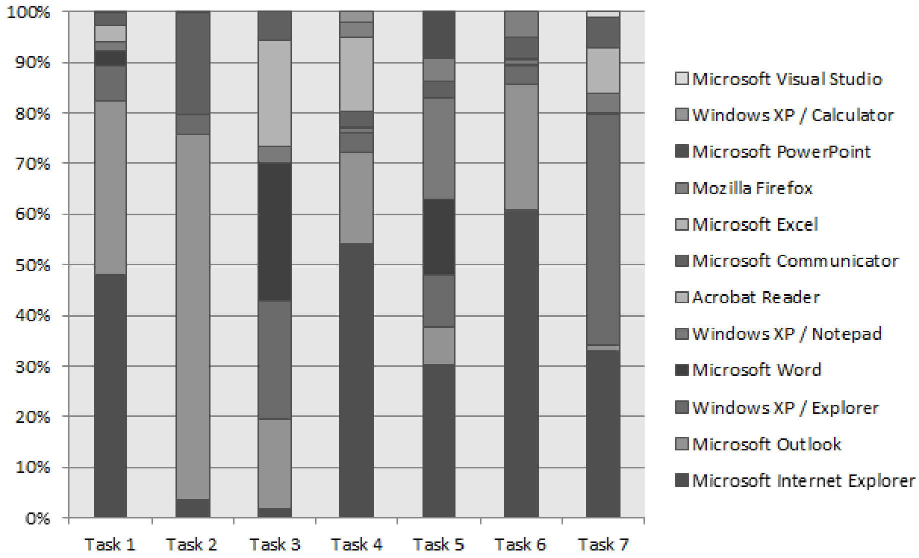

Figure 3.3.: Distribution of application usage per task.

	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7
Mean number of switches	12.85	6.50	33.05	22.70	30.80	10.35	7.50
Mean time between switches	25.32	46.72	16.94	21.51	11.44	42.57	15.47

Table 3.3.: Switch number and duration per task.

connection between the amount of identified segments and the execution duration was identified (see bubble charts 3.5 and 3.6). For the simpler as well as for the more complex tasks, the participants tended to report between four and six segments. It is possible that limitations of the working and the short term memory result in a focus on four connected execution steps. Nevertheless, this aspect requires further investigation in future work. In the following, the segments are investigated further with respect to similarity among participants, repetitions and the mapping to the execution process.

3.3.2.3 Segment Recurrence among Participants

To investigate into recurring work types and the execution process, the study organizers reviewed the identified segments. Next to the experience gained based on the shadowing, they used notes taken during the study and the video material for the segment review. The following recurrences can be reported among the participants:

- **Task1 (15 data sets):** To communicate information about work, the browsing of documents (6), browsing of the web (9) and document authoring (6) were mentioned frequently. Some participants knew the content directly and only reported authoring the text and its communication (via communicator or via email). The participants who reported about the information identification phase fall into two groups. Those that give a detailed overview of the websites they visited and others who only speak of more generic browsing on *interesting* sites.

Task	Average time	Max time	Min time
Task 1	558	1144	228
Task 2	419	791	205
Task 3	1006	1923	417
Task 4	1023	1956	481
Task 5	867	1456	481
Task 6	715	1620	117
Task 7	242	405	76

Table 3.4.: Task execution durations.

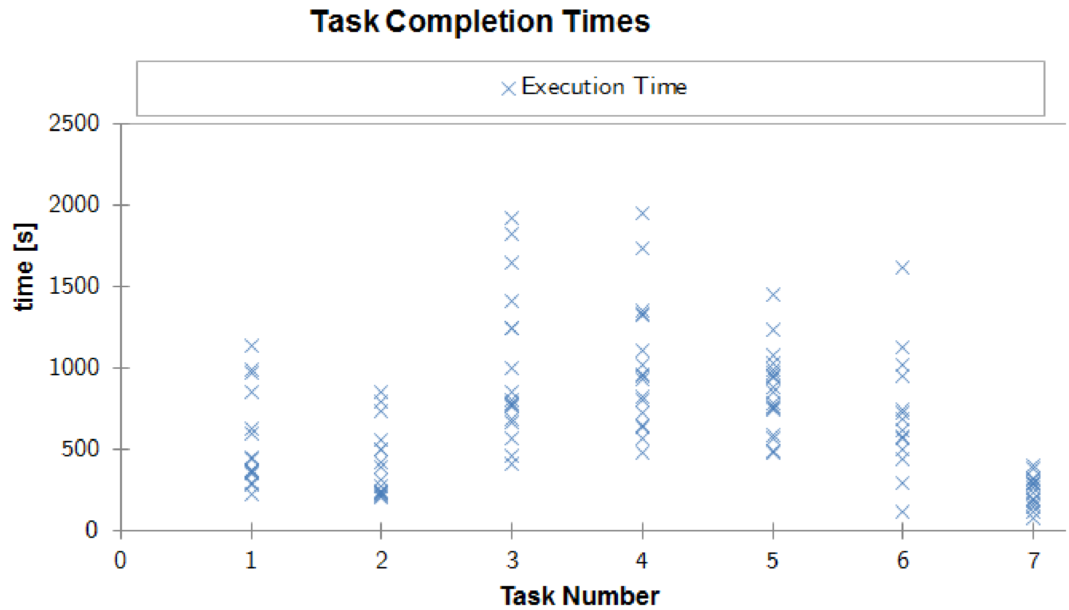

Figure 3.4.: Execution time.

- **Task2 (15 data sets):** Core elements for the meeting creation task was checking the Outlook Calendar (8 times), creation of a meeting (11 times) and creation of a reminder. Some participants reported each check they made (room, timeslot) while others reported on those checks by a single item. Though most participants used Outlook to identify a timeslot, four participants directly used the Windows Messenger to ask for a time slot.

- **Task3 (17 data sets):** To decide on an application the information collection and the dissemination of a decision was mentioned most frequently (16 read application, 11 read job posting, 9 send decision, 5 write mail with decision). Only two mentioned the decision making as part of the process. Few gave detailed information about operations on files (e.g., opening, browsing content).

- **Task4 (18 data sets):** The core process included informing about the conference (mentioned by 10), general web browsing related to the conference (9), searching and booking room and flight (8 room, 9 flight). Cost calculation was frequently mentioned (8) which covered a complex process of data collection, combination and analysis. All processes ended with an information for the manager, e.g., inform about conference (10), send travel request (12), sometimes both combined. Only one participant reported about decision making. One participant delegated the complete task.

- **Task5 (18 data sets):** The translation task was executed very diverse. Some participants could translate the texts by themselves which resulted in statements like understand (1), read doc (6). Others had to use a translation service which involved the search for a service (3) and the use of the service (10). An initial search for the document to be translated was reported by five. Some users did not have the background knowledge for the documents (they reported facts about complex event processing) which resulted in additional web browsing (5). Two asked colleagues for help during the execution. The creation of a presentation was reported differently, based on the chosen format (5 create presentation, 2 generate key facts, 2 send mail).

- **Task6 (14 data sets):** The identification of experts was tackled by most participants by a web search. The search was communicated very differently: general search (6), browse web (6), read Wikipedia (3), search information objects (2) and use tools (1). One participant mentioned think as activity. Refinement, verification, examination were used to describe the organization of the gained information. The communication of the decision was described abstractly (5 communicate) or more specifically (2 send mail, 1 create mail). One participant stated that he delegated the task due to a lack of knowledge.

- **Task7 (18 data sets):** The identification of a shortcut for Visual Studio was handled differently among the participants. Many mentioned that they searched for resources on their local hard drive (11 search file). Some participants used trial and error within the program (4), others browsed the web (4) and one used the visual studio help function. Some participants switched between different strategies if a strategy did not promise quick results. Four participants reported that they stored the shortcut for reuse. The test of the shortcut was only performed and reported by three participants.

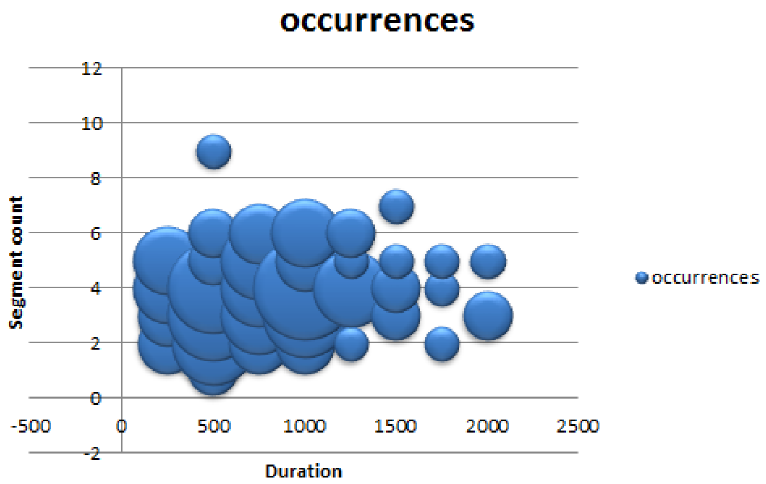

Figure 3.5.: Bubble chart segment count by task execution duration.

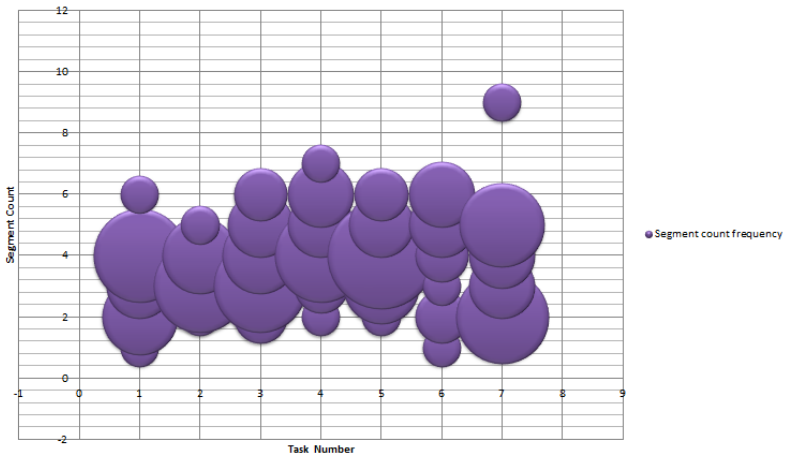

Figure 3.6.: Bubble chart segment count by task number.

For all tasks a difference in the granularity of the segments is obvious. The different granularity emerged although participants were asked to choose a level of detail that was appropriate to give another person an understanding of the execution process. A review of notes and videos supports the idea that a low granularity is chosen for those elements of work the individual was unsure about while experience leads to a less fine granularity. For this thesis, the high overlap shows that different participants chose comparable work processes and described them in a comparable way. This is investigated further in the next section.

3.3.2.4 Segment Characteristics

The segments identified by the users relate to sets of human computer interactions. Based on the reported segmentation, the impression of linear execution processes occurs. Users described their work step by step and did not hint to loops between different steps (e.g., reading document, then authoring etc.). Only one participant hinted to repetition by using loops. A large amount of activity switches exists (see 3.3) while the amount of involved applications never exceeds ten applications. This is an indicator that users visit an application more than once during one task execution process. If the segments are related to applications, then a linear execution is unlikely. A closer investigation in the following shows that the linear description is more the exception than the rule:

- **Application transitions:** To investigate further into the execution process, the application switches have been investigated further. The most frequent transitions between applications have been extracted (see table 3.5). The switches generally occurred between the applications Internet Explorer, Excel, PowerPoint, Adobe Reader and Outlook. With 269 switches, the

	Wind Search	firefox	iexplore	EXCEL	AcroRd	commu- nicator	notepad	Word	OUTLOOK	explorer
Wind. Search	0	0	0	0	0	0	0	0	0	2
firefox	0	2	2	0	0	0	0	0	0	4
iexplore	0	0	0	24	2	0	0	0	36	92
EXCEL	0	0	22	0	0	0	0	0	4	32
POWERPNT	0	0	4	0	0	0	0	0	0	2
AcroRd32	0	0	0	0	0	0	2	12	6	20
communicator	2	0	0	0	2	0	0	0	2	16
calc	0	0	4	8	0	0	0	0	0	4
notepad	0	0	0	0	2	0	0	4	0	4
Word	0	0	0	0	12	0	2	0	14	18
OUTLOOK	4	0	42	2	6	2	0	10	0	44
explorer	4	0	90	32	18	6	8	20	46	0

Table 3.5.: Selection of most frequent transitions between applications (Read: from row application to column application).

application most frequently switched to was Windows Explorer (involving the Windows Desktop, Program Manager, etc.), which includes all types of file operations, such as opening, searching or moving of files. The distribution of the switches shows that between some applications a high switch traffic occurs. An example is Outlook and the Internet Explorer. Users switched 36 times from the Internet Explorer to Outlook and 42 times from Outlook to the Internet Explorer.

- **Sequences with repetitions:** Sequence mining [88] brings more structure into the application switch behavior. Sequence mining is a model-based clustering technique that partitions sequence data according to the order of occurrence. The number of clusters is an input parameter. Each cluster is a first-order Markov chain that is capable of reproducing a set of sequences associated with the respective cluster [282]. The application switch sequence was input to the sequence mining.

  The sequence mining method generated one Markov chain per task. It is a graph with nodes, representing applications with application relevance and edges, representing node transitions (cluster number was set to one because tests showed that more clusters did not improve the insight). Based on the probabilities and relevance, irrelevant applications and transitions were excluded. The resulting Markov chain represents a core execution process, including the relevant elements of work execution and the respective transition probabilities.

  Figure 3.7 shows the resulting Markov chains. Highly relevant nodes (most time spent using the respective applications) are generally connected with high transition probability. The connections stand for application switch cycles that occur when information is transferred between two applications. For example task 2 is executed, using Outlook and Internet Explorer with users starting generally by using Internet Explorer. Once users switch from Outlook to another application there is a high probability that the Internet Explorer is used.

- **Sequences and segments:** The study organizers mapped applications to segments based on the understanding they gained while they shadowed the participants. The focus were the most frequently identified segments and the most frequently used applications. The Outlook note was related to send mail and communication activities (similarly the communicator node). The Internet Explorer and Firefox nodes were related to segments that tackle web browsing, web searching and access of specific websites. The Notepad, Word, PowerPoint node were related to text production and consumption activities (except writing mail). The pdf viewer was related to text consumption. Excel and the calculator were used for calculation activities.

  As a result, the mapping shows that those nodes the users switched frequently between are often related to different segments. The linear reporting of the work process covers a distributed, repetitive execution of activities that are closely combined in the task execution process.

The three step investigation into the structure of the work execution (involving application transitions, sequence mining and a mapping of frequently used segmentation to the mined sequences) has shown that the described segments are not executed linearly. The tendency to a linear description might be related to a progress-focused individual understanding of work execution, tending to report like a narration. In fact, information work presents itself more as a circular, repetitive movement along applications and information objects while generating progress. The cognitive steps seem to be limited to few segments. A subset of the reported segments repeats among the study participants. At the same time, the segments repeat during the task execution process in the sense that they are interwoven and the progress of each segment seems to depend on progress in another segment.

3.3.2.5 Recurring Units of Work

A major goal of this section is the identification of recurring, logical units of information work which result from the application of work techniques. While the review of existing taxonomies enforced the assumption of existing logical units of work, consistency issues discouraged the use of an existing taxonomy within this thesis. The conducted study provides further evidence for the presence of logical units of work due similarities within the execution processes among the participants.

The idea is that each information worker has an implicit understanding of the logical units of work. The challenge is to extract this knowledge. In the following, the idea is to extract this implicit knowledge based on the study participants' renarrations of the work process collected in the questionnaires. Zacks et al. describe that humans apply taxonomic and partonomic structures when they perceive and describe events [302] (partonomic = "part-of" relation, taxonomic ="is-a" relation). For the terms used to renarrate the work process one can assume that the participants classified the renarration based on their personal taxonomies work techniques which generated the described units of work. The extraction of the logical units of work is a challenge as individuals—even if they talk about the same work technique—might use different vocabulary. Furthermore, the logical units of work identified based on the questionnaire need to be validated against the collected interaction logs to assure that the units of work reference actual interactions clearly and without overlap.

The following process was applied to extract the logical units of information work from the collected data:

- **Extraction of logical units of work from questionnaires:**
  - *Process:* For each task that has been executed within the study, the questionnaire asks for a renarration of the work process based on a provided structure (e.g., "I *browsed detailed job offers* WITH THE GOAL/TO *examine applicants*", cf. section 3.3.2.1). The interesting element are the verbs of this structure. The verbs describe the performed activities and need to be extracted.
  - *Result:* Overall, 30 different verbs have been extracted which have been used by the participants to describe their work process. The verb list shows that many verbs have been used very frequently (see left column in the table 3.6). Verbs especially have not been used only by one person but the extracted data indicates that a shared vocabulary for work execution exists. Nevertheless, as the vocabulary is not standardized some verbs obviously are synonyms. Therefore, a unitization of the verbs is required.
- **Unitization of vocabulary and taxonomy extraction:**
  - *Process:* To remove synonyms from the verb list the following process was applied. Within the focus group which created the study tasks card sorting was conducted to remove synonyms and thus remove redundant identifiers for similar units of work (for details about the card sorting technique, see [61]). The verbs extracted from the questionnaires were written on cards. The participants were informed that the verbs stand for work techniques. Each participant received a complete set of cards containing all verbs and was asked to cluster those verbs which refer to similar work techniques. This first clustering only served the purpose of familiarizing the participants with the task. Afterwards the card sorting was repeated in the group to generate a sorting supported by all participants. As a result different clusters of verbs resulted, containing verbs which were considered to address similar work techniques. For each cluster, the group discussed which verb described the work technique best.
  - *Result:* The familiarization of the group members with the vocabulary by a card sorting on ones own was considered beneficial by the group members. The card sorting conducted by the group quickly (roughly 15 minutes) generated a sorting supported by the whole group. Overall, 12 different groups were created. Discussion emerged for very generic terms like execute but the group was able to generate consensus for all discussions.

    During the sorting, the group members frequently started to talk about the granularity. Initially, they expressed the feeling that the granularity of the units of work referenced by the verbs was very different. Discussions in the group found the agreement that the verbs actually addressed two different granularity levels. Therefore, the participants began to separate the clusters into two groups. One group containing complex units of work (e.g., authoring, searching) and another group which contains fine grained units of work which describe actual interaction with the computer (e.g., opening, saving). Finally, a work technique list was developed (see Table 3.6).
- **Sample based evaluation of identified units of work:**
  - *Process:* The described process identified units of work solely based on the study participants' renarration of the task execution processes. The verb list generated based on the card sorting process in the focus group seems to identify work techniques. Open questions remain: Are the identified work techniques really resulting in similar logical units of work, executed similarly by different participants? Do participants use the comparable execution processes for other types of work?

    To address these questions a sample based analysis of the interaction logs was conducted. A set of 3 verbs from the unit of work list was chosen randomly. The considered verbs were authoring, browse and open. The questionnaires which used the verb (or which used a term that belongs to the cluster, the verb in the unit of work list originated from) were identified and the respective user interaction records were accessed. The study organizers analyzed the interaction process and identified the interactions, the respective verb referred to (XML files with the interaction sequence based on monitored interaction events). This was done by the two organizers together and doubts were directly addressed by discussions. As the time distance between the data collection and the analysis was 2 weeks, the organizers remembered different facts from the executions which supported the analysis process.

If the applications or the used functionalities the unit of work referred to were similar among the participants' data sets, the assumption that the verb with the respective unit of work is useful was not rejected. Additionally, other occurrences of the respective interactions were searched within the interaction logs (full text search in XML files). For the identified occurrences the questionnaires were used to check whether the respective interactions were addressed by different units of work. If no other occurrence was addressed by a different unit of work, the assumption that the verb as logical unit of work is useful was not rejected as well.

- *Result:*

  The analysis was complex due to the circular movement between applications during the execution processes. Nevertheless, the organizers were confident to have assigned the correct applications and functionalities to the considered units of work. The analysis showed an overall usefulness of the identified units of work because the two questions initially mentioned were verified: 1) Participants used applications with similar functionalities for the respective units of work and used comparable functionalities (e.g., authoring was performed using tools like Microsoft Word and the Notepad application together with events of type typing and using functionalities to modify the text style). 2) If similar applications and functionalities were used elsewhere in the interaction logs, the participants did not assign different units of work to the respective interactions.

  An interesting finding is that many occurrences of work techniques were not included in the renarrations. Especially the fine grained work technique open was only mentioned in very few cases while it was performed very frequently according to the interaction log. This shows that renarrations are an important source of information while it must be considered that they only contain those aspects study participants consider relevant.

  The open work technique occurred frequently while the overall work technique was browsing. On the other hand, the browsing unit of work and the authoring unit of work shared the text input interaction. Nevertheless, browsing and authoring did not overlap as the complete set of functionalities and application allowed a clear distinction. The assumption is that the broader units of work contain the smaller units of work in a partonomic relation and that sets of functions and applications can be used to reason about units of work based on identified interactions.

  Overall, the interaction log based review of the identified work techniques showed that the techniques address similar interactions and that they do not overlap with other units of work of the same granularity. Thus, the work techniques are of practical use when it comes to the description of work processes at the computer. Indication for a partonomic relation between the different granularities have been found.

- **Harmonization with existing theoretical approaches:**
  - *Process:* A list of work techniques has been identified and the applicability of the techniques has been validated by analyzing a sample of units of work based on the logged interactions. In the following, the units of work are organized and set in relation to the reviewed literature (see section 3.3.1).
  - *Result:*
    * PARTONOMIC RELATION: Based on the verbs extracted from the questionnaire, the focus group has identified two different work technique types. The first enclosing group contains verbs of low granularity that match software functionalities: users use words that belong to the desktop metaphor and the WIMP user interface paradigm (windows, icons, menus, pointers) [279] and enumerate system functionalities (e.g., open a document, enter a website url, etc.). The second enclosing group describes more complex work techniques which require a large set of interactions (e.g., authoring, searching).

      The investigation into the interaction logs provided evidence for a partonomic relation between the two groups. It is assumed that the complex work techniques combine the simple techniques to compose interaction processes.
    * ACTIVITY THEORY VOCABULARY From an AT point of view the identified units of work which result from work technique application are activities. The complex work techniques require different interactions to achieve a goal. For example the authoring of a document requires the coordination of many typing activities and complex cognitive processes which organize the language to be written down. The complexity resembles the action level of the activity hierarchy (in some cases it might even be an activity on its own). The fine grained work techniques on the other hand do not require complex mental effort. Opening a file triggers a clear and each time

Action specified	Occurrences	Operations (low granularity)	Actions (high granularity)
Search	65		Browse
Browse	57		Browse
Read	41		Consume
Open/Access	34	Open	
Author/Write	31		Author
Create	28	Create	
Communicate	22		Communicate
Send	20		Communicate
Use	16	Focus	
Copy	13	Copy	
Save	11	Save	
Organize	10		Organize
Examine	9		Consume
Calculate	8	Execute	
Execute	5	Execute	

Table 3.6.: Work techniques: Clusters identified for verbs used in the task segmentation.

similar interaction process (as long as the subject knows where to find the file). In terms of the activity hierarchy, the resulting units of work resemble operations (i.e., the subject is able to internalize them completely).

An AT based perspective on work techniques has already been taken by Hädrich who described knowledge actions as well [109]. Hädrich's taxonomy was not used for the further work in this dissertation as it was not directly applicable for the computer workplace. Hädrich develops knowledge actions as work techniques which are applied within information work execution processes. Furthermore, Hädrich specifies workflows which realize knowledge actions. A relation similar to the partonomic relation between the complex and the fine grained work techniques identified, here.

In the following, the complex work techniques which have been specified to be on the action level of the AT hierarchy are referred to as knowledge actions. Knowledge actions are considered as abstract work techniques the subject learns and applies to execute complex information work activities. The fine grained work techniques which have been specified to be on the operation level of the AT hierarchy are referred to as desktop operations. Desktop operations are operations which are executed on the computer workplace and follow the WIMP vocabulary.

The identified work techniques 1) focus on information work at the computer workplace 2) can be combined to describe information work execution 3) provide explanation for the actual interactions that can be observed. Thus, the initially identified requirements are met. In the following, the identified units of work—knowledge actions and desktop operations—are presented in detail.

3.3.3 Information Work Unit Taxonomy

The identified work techniques are the foundation for a taxonomy of logical units of work relevant for information work. The taxonomy is presented in the following.

Based on the identified concepts, informal interviews were conducted with the study participants. After an explanation of the idea of work techniques, the identified concepts were provided to the participants. The participants were asked whether they consider the identified elements to be work techniques which resulted in the units of work they considered their work process to be composed of. Additionally, the participants were asked to elaborate on the concepts and to identify additional elements, not captured, yet. The interviews especially helped to get a better understanding of the differences between desktop operations and knowledge actions. The taxonomy and the details presented in the following, thus synthesize the results of the empirical study and the conducted interviews.

3.3.3.1 Desktop Operations

A desktop operation is a user system interaction that relies on trained knowledge about the user interface. Computer systems apply similar desktop metaphors, use the WIMP paradigm and follow design guidelines to realize similar structured applications [279]. Therefore, desktop operations are closely related to the interpretation of the desktop metaphor in the operating system and an interface logic that follows design guidelines. The term operation has been chosen to indicate that an internalization of this type of interaction is likely. Desktop operations represent an operation and therewith a habitual, subconscious routine if the action has been executed

Opr \ Obj	App	File	Folder	Information Object	Window
Open	x	x	x		
Close	x	x	x		
Save		x			
Rename		x	x		
Delete		x	x	x	
Cut		x	x	x	
Paste		x	x	x	
Send		x		x	
Create		x	x	x	
Execute	x				
Focus			x		x

Table 3.7.: Matching of operations and objects for desktop operations (Opr=Operation, Obj=Object, App=Application).

multiple times. It can also represent an action within activity theory if the user has a low level of experience with, for example, a new application and needs to plan his actions in order to reach the desired result.

The desktop operations provided in the following are based on the data provided by the user study. However, the evaluation based set of desktop operations has been was extended to cover other frequent operations not captured in the study due to the limited set of tasks/to cover operations not mentioned by users at all. The focusing of an application window for example happened with each switch but was not reported as being part of the relevant work process. Missing such important aspects of work when segmenting the work execution underpins the assumption that desktop operations require little cognitive effort.

The following desktop operations are considered:

- **Opening:** Trigger the visualization of an object.
- **Closing:** End the visualization of an object.
- **Saving:** Persisting an object.
- **Renaming:** Changing the descriptor of an object.
- **Deleting:** Removing an object from a logical address.
- **Cutting:** Removing an object from a storage or logic location to put at a logical address.
- **Copying:** Creating a duplicate of an object to be put at a logical address.
- **Pasting:** Putting a formerly copied or cut element at a logical address.
- **Printing:** Triggering the creation of a paper-based physical representation of an object.
- **Creating:** Creating a new object which follows and setting its encoding format.
- **Executing:** Triggering the interpretation or execution of an interpretable or executable object.
- **Focusing:** Highlighting an object.

Desktop operations are performed on objects. The following objects are considered to be target of a desktop operation:

- **Application:** An application is a piece of software that can be run in the environment and is used to transform information. A process is a running instance of an application.
- **File:** A file is a single resource of information. Files are accessed, using applications and are stored on a drive.
- **Folder:** A folder is used to organize files. It can encapsulate an arbitrary number of files.
- **Information Object:** Information objects are the smallest possible units of information. This includes, for example, textual information represented by a chain of characters. They can be stored within a file.
- **Window:** A window is the visual representation of content on the computer desktop. It can be associated to an application or to a folder and can show the information objects contained in a file.

On every object, the user can execute a number of different actions but not every operation is possible on every object. Table 3.7 shows the possible pairs which represent the set of desktop operations. One desktop operation is the pair “Opening” – “Application”. It represents opening a new running-instance of an application. The operation “Opening” can also be applied to the object “File”. This signals that the user is opening a file to access its contents. The operation “Focusing” is not a directly observable user-system interaction because it is not actively executed. Rather, we are able to observe the operation which leads to a change of the focused window. Therefore, this layer adds semantic value to the observed activities on the computer desktop. “Focusing” is the only operation which can be used on the object “Window”.

3.3.3.2 Knowledge Actions

The study has identified knowledge actions as complex work techniques. A knowledge action is considered as a solution strategy which results in complex units of words which involve multiple desktop operations. The task execution study has shown that participants combined different knowledge actions to realize a task. A knowledge action used in an activity is an interpretation of the work technique based on the context and the constraints of the activity. The knowledge action as such is the abstract concept of a work technique or solution strategy that can be applied in many different work situations. Five different knowledge actions have been identified for work execution at the computer workplace:

1. **Authoring:** The knowledge action authoring relates to the creation of textual or other media content with the help of technology. Authoring does not refer to the actual purpose of a text product. Authoring is embodied in the existing sets of knowledge actions but it is partially paraphrased with other terms. For example Davenport [68] mentions the documentation of knowledge for its later reuse, which is clearly related to the externalization of one's knowledge in exchangeable artifacts. Sellen and Harper [252] on the other hand call an important activity 'composing' and Völkel [290] calls it 'document creation'. Hädrich [109] explicitly mentions the activities 'authoring' and 'co-authoring', while the latter could also be interpreted as 'document creation' in Völkel's wording.

2. **Communicating:** Communication refers to exchanging or spreading information or information objects. Such an activity can be found in most of the literature examined although also here the authors use various terms to describe the action. Davenport [70] is the only author who uses the term dissemination, while others use communication [72, 262], exposure [81], conveying information [22], emission [123] or sharing and presenting [26] for the same action.

3. **Organization:** Existing information objects are organized by applying an organization scheme or structure. An example is the tagging of documents or the copying of files into specific folders. Also, the knowledge action of information organization looms in the existing knowledge work literature, e.g., [252, 26] explicitly name organizing knowledge a relevant activity. Efimova's [81] view on information work is shaped by the usage of weblogs as tools but she, nevertheless, mentions the organization of ideas as a relevant and weblog-supported knowledge action.

4. **Browsing:** Browsing means looking up information on a specific topic or problem in a specific form. Personal, organizational file storages or the web are subject to browsing. Information retrieval services such as search engines support the browsing process. Related actions can be found in the works of [22, 199, 26] and is also discussed in a broader meaning in [68, 262, 252].

5. **Consuming:** The information worker focuses a resource on the computer desktop and processes the visual representation of the underlying knowledge. The reviewed literature relates to consumption by terms like review [72], assimilation [123] or evaluation [26]. The term consumption has the advantage that no goal beyond the consumption of information is described. Consumption will have specific goals dependent on the task it is embedded in but which is not easy to discover.

The given list of knowledge actions focuses on the execution of computer work. The used abstraction is chosen to avoid assumptions about the goal of an action beyond the observable production, transformation or perception of an element. This limitation follows the idea that work techniques are agnostic to a goal until they are actually used in the context of an activity. Therefore, the technique has no goal of its own beyond the interactions that realize it as logical unit of work.

3.3.4 Intermediate Results

The section has identified knowledge actions and desktop operations as recurring elements of information work at the computer workplace. A literature review has identified existing lists of recurring elements of information work (see section 3.3.1). The literature discusses recurring elements without specifying the granularity and without avoiding an overlap of elements. Therefore, and to focus more on the computer workplace an exploratory study was conducted (see section 3.3.2). The study analyzed the execution processes of seven knowledge intensive tasks and asked for a segmentation of those work processes to classify the segments.

The data analysis and the processing of the data resulted in an understanding of information work as a circular, repetitive movement along applications and information objects while generating progress. Actions repeat during the task execution process in the sense that they are interwoven and the progress of each action seems to depend on progress in another action.

Broad- and fine-grained types of interactions were identified. Fine-grained elements are termed desktop operations. Desktop operations stand for direct human computer interactions on a functional level. Operations are assumed to be quickly internalized and to be executed without much mental effort. The broad-grained elements are termed knowledge actions. Knowledge actions are work techniques or solution procedures that can be applied and combined for different tasks. All tasks of the study were executed by a combination of different knowledge actions. Each knowledge action contained a set of desktop operations.

A taxonomy of knowledge actions and desktop operations has been specified (see section 3.3.3). The knowledge actions focus on observable activities of information production, transformation and perception. For the desktop operations a list of compatible interface elements relevant for desktop-centric operating system environments was created.

Basic characteristics	
Effectiveness	Many goals are identified by the subject. Regulation influences the acceptance criteria of those goals. As a result effectiveness can be achieved by regulations even if the original goal is not achieved.
Efficiency	Efficiency depends on elaborate operational cognitive images.
Information	Information is product and raw material of information work.
Computer workplace	The computer workplace structures the work process. Due to multi-tasking capabilities work on multiple activities in parallel is facilitated.
Information work coordination	Self interruptions and external interruptions are used to coordinate work processes.
Information work techniques	Information work execution applies trained work techniques to address complex and underspecified goals. Knowledge actions are combined in cyclic work processes to realize those complex goals. Knowledge actions trigger simple interaction techniques to operate the computer which have been termed desktop operations.

Table 3.8.: Information work ideal type.

Method Design	
Information overload	Large amounts of information complicate the data organization and may result in a loss of overview.
Complexity of the workspace	Many complex applications are combined in work processes. Wrong tool choices may decrease the efficiency of work.
Underspecified work process	The subject needs to identify a goal directed work process. Without elaborate operational cognitive images the work processes can be inefficient.
Interruption based coordination	Interruption is a core element of the work process to realize the coordination among different goals. Nevertheless, interruptions are also the cause of memory failures.

Table 3.9.: Information work threats.

3.4 Summary

This chapter has outlined the information work ideal type. The ideal type provides a unified analytical framework for information work execution. The ideal type has been developed in close alignment with AT and ART. The focus is on the activities performed within information work, the cognitive processes they involve and the objects they are performed on. The focus of the ideal type is information work at the computer workplace.

Information work has been identified as a product of the information society to address control problems in a global process of transformed commodity and information exchange (cf. section 2.4). A high degree of autonomy results. Information plays a crucial role to establish responsive productive processes. The ideal type specifies how these characteristics translate into work processes. Four basic aspects have specific relevance for the course of this dissertation:

- **Underspecification of work, effectiveness and efficiency:** The information worker has to identify work processes to accomplish underspecified goals. As an effect, regulation is frequently applied to assure the goal-oriented execution of activities. Effectiveness and efficiency are heavily influenced by the execution of underspecified goals because the acceptance criteria for work as well as the identification of failures are frequently generated by the subject himself (see section 3.1).

- **Interruption as coordination:** Interruptions have a coordinative role for the work execution process. While they are relevant to ensure the successful work execution they are directly related to prospective and retrospective memory failures (see section 3.2).

- **Logical units of work:** Knowledge actions and desktop operations have been identified as building blocks of information work. Knowledge actions are work techniques applied by information workers to address new problem domains based on well-known work procedures. Knowledge actions are composed of desktop operations. Desktop operations are the basic categories of interaction used to interact with the computer (see section 3.3).

The ideal type enables modeling and analyzing information work execution. For example the creation of a report can be described in terms of the knowledge actions applied (consumption and authoring). The knowledge actions trigger desktop operations like starting applications, entering terms and consuming services. The cyclic process of searching for information and persisting it in the document is covered by the ideal type as well as the reaction to interruptions. If a colleague enters the office while the report is constructed, the subject needs to switch the activity to interact with the colleague. The switch involves the memorization of the latest activities. If the colleague provides information which requires a modification of the report, the conversation with the colleague and the report creation merge and the subject applies regulation. The subject reasons about the effectiveness based on personal criteria of a good report unless report quality specifications exist.

An overview of the ideal type is given in Table 3.8. The specification of the ideal type already contains several threats which complicate information work execution. It is necessary to consider these threats later within this dissertation (which is done in the system design phase, see chapter 5). To give an overview, the threats are summarized in Table 3.9.

The basic example shows that rich models of information work can be constructed based on the ideal type. Especially unforeseen elements like external interruptions, goal modification and process regulation are covered. The ideal type is a suitable foundation to build models of information work and systematically analyze memory threats and support capabilities. The next part of this thesis addresses this analysis and the respective modeling.

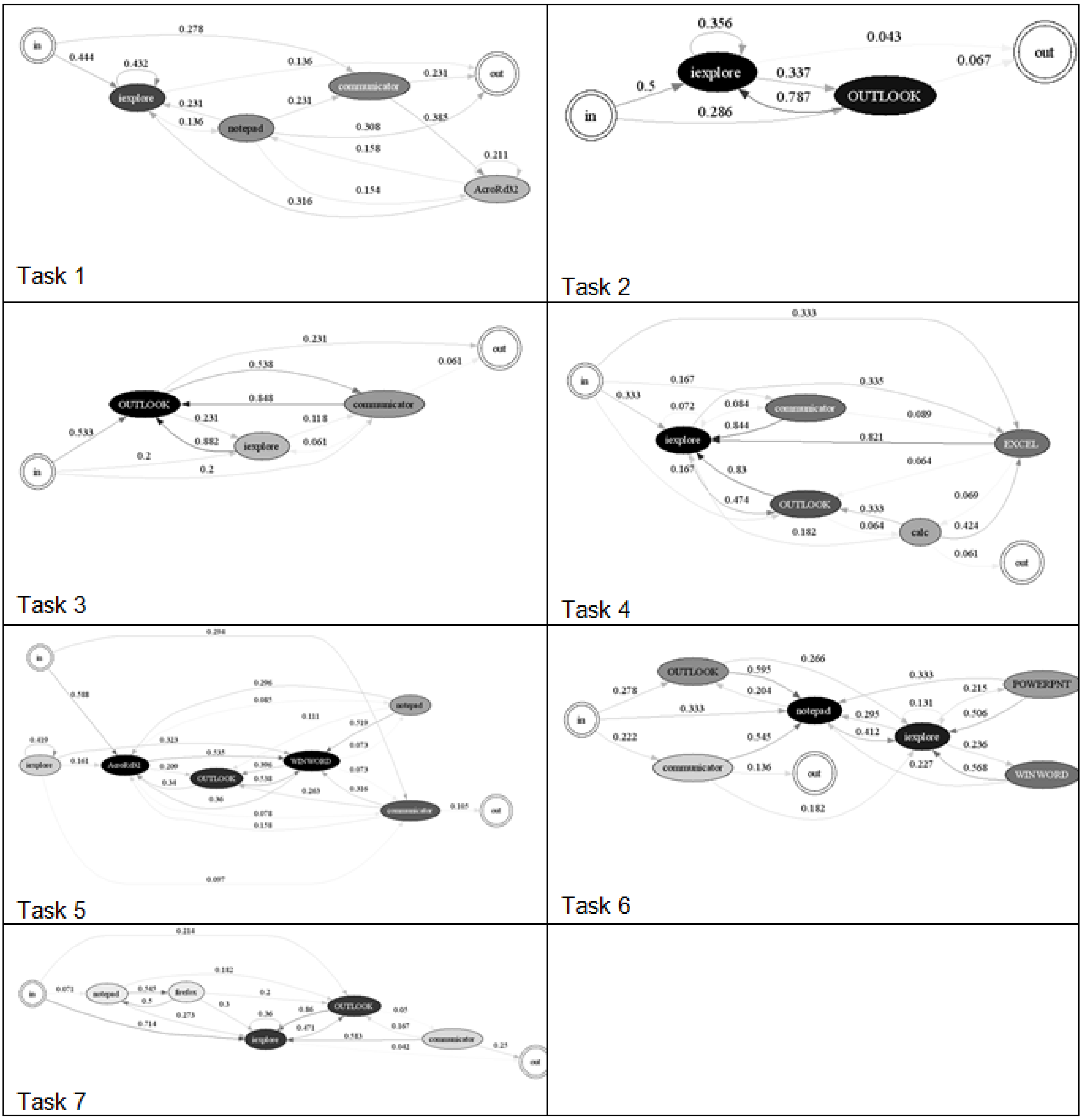

Figure 3.7.: Markov Chains for task execution processes (the darker a node, the more time was spent using the node/the numbers at the edges denote the transition probabilities).

Part II.

System Design for Information Work

4 System Design Method for Information Work

This chapter provides a system design method to create a system that addresses memory threats in information work at the computer workplace. System design methods describe frameworks for software development as engineering discipline [12]. A system design method for the domain tackled here needs to consider the multitasking nature of information work and the dynamicity of the work process. Characteristics which are closely related to cognitive processes of work coordination. Therefore, a system design method is required which considers the cognitive processes as well as a subject's actual interactions with the world involved in information work execution.

As an overall system design direction, user-centered design (UCD) is chosen due to the strict focus on user requirements (see section 4.1) [171]. A limitation of UCD is the lack of appropriate methods to model cognitive processes involved in work execution. This chapter proposes the activity theory based system design method (AT-SDM), a method set to model cognitive processes in the context of use analysis and the requirement specification of the UCD. First, the basic elements of the AT-SDM are provided which integrate cognitive activities and interaction activities in one model (see section 4.2). The model is based on the principles of activity-theory (AT) and action regulation theory (ART). The model is used within the AT-SDM to analyze the coordination of work processes and the respective interactions. Therefore, model properties are defined which are required to specify coordinative processes like activity switches (see section 4.3). The properties also enable the analysis of the model to identify problems within the work domain specified by the model (see section 4.4). Finally, the application of the AT-SDM within UCD is described (see section 4.5).

4.1 User-centred Design to Develop Information Work Support

This section decides for a system design method used within this dissertation. The system design method needs to be appropriate to analyze information work. This means that the attainment of multiple goals based on various activities that are coordinated by cognitive processes needs to be analyzed by the system design method.

A short investigation into system design methods comes to the conclusion that user-centered design (UCD) is an appropriate system design method to tackle information work (see section 4.1.1). The basic process of UCD is presented (see section 4.1.2) and the applicability of UCD to address the specific challenges of information work analysis is investigated in detail (see section 4.1.3).

4.1.1 Benefits of User-centred Design

A plethora of different system design paradigms exists. Examples are the UCD cycle [198], waterfall models [5] and spiral models [29] to name only a few. Each paradigm defines an execution process of sequentially or cyclic connected activities. Each process step can be realized by different methods. Therefore, the application of system design demands the selection of a paradigm as well as the selection of applied methods. Examples for methods of requirement engineering that can be applied for the waterfall or the spiral model are i* [300] or object oriented analysis using UML or SysML [93].

Waterfall models and spiral models realize a straight forward process of requirement engineering and provide means to coordinate complex development processes. The UCD system design method focuses on the identification and addressing of user requirements as complex challenge. This is achieved based on three basic principles [171, 170].

- **Iteration:** An iterative approach assures that requirements and designs are refined until a suitable solution is developed.
- **Participation:** During all iterations the participation of end users is stipulated.
- **Distribution:** The distribution of tasks between human and machine is a basic principle of UCD.

Due to the focus and the respective principles the use of UCD for a system design in the domain of information work is beneficial. Another argument is even more important. The requirement engineering for a support method which addresses memory failures is no straight forward process. Due to the involved cognitive processes which are not directly observable a complex process of balancing requirements can be assumed. The iterative structure of UCD directly addresses this process of balancing requirements and designs.

To conclude, UCD is considered to be an appropriate method for the system design conducted within this dissertation.

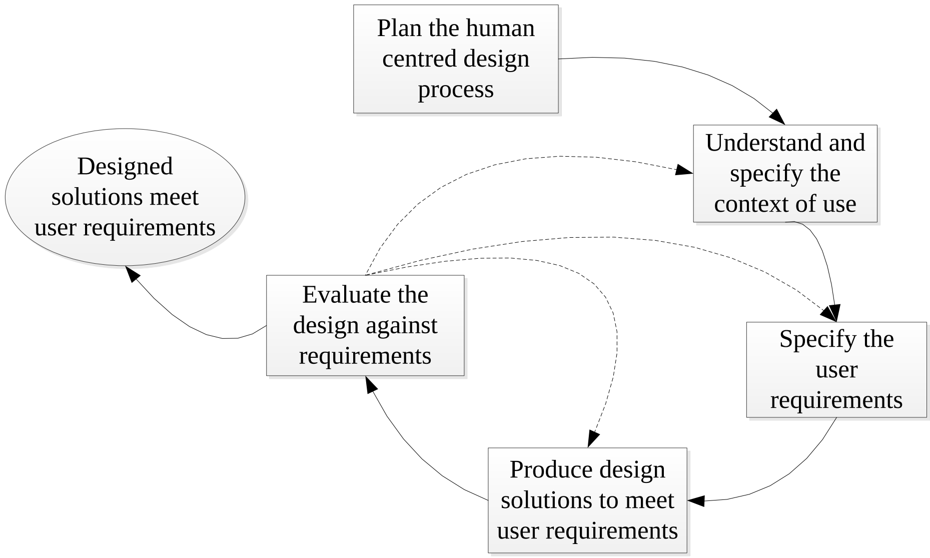

Figure 4.1.: The user-centered design cycle as defined in [136].

4.1.2 The User-centred Design Process

In the following, user-centered design is investigated further. The term UCD was originally termed in 1986 by Norman [198] for a cyclic system design process with a focus on user involvement in the design process. Nowadays, UCD has become increasingly popular both in academia and industry. This resulted in the creation of the ISO standard [136] for UCD.

The UCD cycle (see Figure 4.1) comprises five steps, summarized in [171]:

- **Plan the UCD:** The planning includes the decision on an overall goal, the intended users and known technological constraints. Applicable methods are usability planning or the usability cost-benefit analysis.

- **Understand and specify context of use:** The context of use is a representation of the state of knowledge that exists about the application domain. Therefore, the collection of information regarding user groups, tasks, technical, physical and organizational environments is necessary to specify the context of use [170]. Applicable methods to specify the context of use analysis comprise interviews, diary keeping, user observation, surveys and task analysis [73, 62].

- **Specify the user requirements:** Based on the context of use and the initially defined scope requirement engineering is performed, involving requirement analysis, requirement elicitation and requirement specification. UCD is often realized with techniques like scenarios of use [270], personas [60] and user stories [58] to structure the requirement engineering process.

- **Produce design solutions to meet user requirements:** A process of design creation and prototype development addresses the identified requirements. Typically rapid prototyping [284] or rapid application development [171] are used to realize this.

- **Evaluate design against requirements:** Empirical methods are applied for formative and summative testing. Formative testing addresses the improvement of the product as part of the development process. Summative testing tests whether the concept helps to successfully address the requirements. Dependent on the test results, the UCD cycle is completed or an earlier stage of the UCD cycle is repeated.

For a system design based on UCD appropriate methods for the execution of the different steps need to be identified. Until now no common guidelines for the method design have emerged. As a consequence method selection is a complex process because chosen methods are not necessarily compatible (e.g., Viller notes that task analysis methods that are not compatible with rapid prototyping [284]). Another challenge related to the method selection is the transition from one step to the next step. Each step builds on the results of the previous step. Therefore, the method of the previous step needs to create results in a form which can be directly used in the next step. Examples are methods which facilitate the transition between context of use analysis and requirement specification are provided: diaries for the context of use + scenarios for requirement specification, personas for the context of use + user stories for requirement specification.

4.1.3 User-centred Design and Information Work

For the domain of information work, the selection of an appropriate method set for UCD is complex. The methods to realize UCD for information work at the computer workplace need to consider its characteristics, especially multitasking and the dynamic work processes (cf. chapter 3). Characteristics which require a consideration of the mental processes which coordinate the task switches and the work process. Only based on the cognitive processes peculiarities like memory failures can be considered in the system design. To consider these processes within an application of UCD, the context of use specification and the requirement specification are of specific relevance. For both steps methods need to be identified which consider the cognitive processes involved in work execution.

Methods to realize the context of use analysis and the requirement specification are reviewed in the following. The methods have not necessarily been developed for UCD but there is no objection of using them to realize respective steps within the UCD system design method. The review focuses on methods which are based on activity-theory (AT). Due to the broad range of specific methods a decision for methods to be reviewed was required. The decision was taken in favor of methods based on AT because cognitive processes and work decomposition are inherent to AT (cf. section 2.1). An even stronger trend towards a consideration of the cognitive processes was expected from methods based on action regulation theory (ART) (cf. section 2.2). However, it was not possible to find any method in the tradition of ART.

The review assesses the methods with respect to the integration of cognitive processes in the context of use analysis and the requirement specification. Another relevant aspect is the transition complexity between a context of use and the requirement specification.

- **Ethnographic context-of use identification:** Ethnographic methods are used to specify the context of use. The activity checklist is a survey to support the context of use specification [142]. The list is used to support interview processes that are intended to identify relevant context elements. Another approach is activity analysis. Activity analysis is a method that applies ethnographic field studies and an analysis of observations to specify the context of use [21]. Activities within a setting are identified and, in a second step, patterns of these activities are analyzed. Overall, ethnographic methods provide a context of use which considers the object orientedness of a subject. Object orientedness addresses cognitive processes involved in the execution of a single activity. The cognitive processes to coordinate between different activities are not covered. A requirement specification method appropriate to the context of use generated by ethnographic studies is not specified.

- **System creation for context of use analysis and requirement specification:** Different methods build on Engeström's system model to create a context representation as systemic relations. Neto et al. [195] combine the i* framework for organizational modeling and AT to address context of use analysis. Martins and Daltrini [80] use Engeström and decompose identified activities into actions and operations. Based on the creation of the systemic relations and the hierarchical decomposition of activities, requirements are elicited. Cognitive processes are considered with respect to the decomposition of activities into actions and operations. However, the coordination between different activities and the ad-hoc creation of work processes is not considered.

- **Tension analysis:** Other methods use Engeström's model and explicitly integrate the identification of tensions in the designed model. Collins et al. [59] show the applicability of AT to capture data collected in interviews. A hierarchical implementation of Engeström's model is used to organize interview findings and to identify tensions to derive design requirements. The tension analysis methods are appropriate to identify requirements without complete activity knowledge. The investigation into relations between different activities, as it is given with multitasking, is not included in the methods.

The review of the methods has shown that AT based methods consider cognitive processes involved in work execution to a certain degree. Nevertheless, the cognitive processes to coordinate between multiple activities and to create ad-hoc work processes based on those decisions are not provided. An interesting insight of the review is the usefulness of activity system model (ASM) analysis to identify characteristics of activities (tension analysis methods).

In the background chapter (chapter 2) the limited explanation of cognitive processes was the reason for the introduction of ART. The following section will continue this path. Concepts from AT and ART will be combined to realize a method set which covers the context of use analysis and the requirement specification. The method considers the cognitive processes which coordinate multiple activities and generate the respective work process.

4.1.4 Intermediate Results

The taken decision for UCD structures the whole design process reported within the next chapters of this dissertation. However, before the domain can be analyzed appropriately a method is required which realizes a context of use analysis and the requirement specification under consideration of the involved cognitive processes and interaction with the world in information work processes. Only based on such a method a decent analysis of the context of use is feasible due to the relevance of the cognitive processes for work coordination. The remainder of this chapter specifies an appropriate method.

4.2 Activity Theory based System Design Method

This section introduces the activity theory based system design method (AT-SDM), a method to perform a context of use analysis and a requirement specification for a domain. One goal of the AT-SDM is to model cognitive processes which coordinate multiple activities and generate the respective work process. The AT-SDM develops a system model which stands for the context of use of the analyzed domain. Different analysis techniques exist which can be used to identify threats within system models and to identify requirements. This section focuses on the development of the system model, thus covering the context of use analysis. The system model has two core elements: ASMs and the heterarchy. ASMs are used to model the systemic relations of the elements involved in an activity (see section 4.2.1). The system models cover cognitive activities as well as actual interactions with the world and are based on the system model approach introduced earlier (see section 2.1.3). The heterarchy connects ASMs. The resulting structure of connected ASMs is used to model the relation between cognitive processes and the triggered interaction activities (see section 4.2.2). The use of heterarchies closely follows their application within ART (cf. section 2.2).

4.2.1 Activity System Constructs

A context of use based on the activity theory based system design method (AT-SDM) models all separable goals or objectives identifiable in the considered domain based on activity system models (ASMs). The ASM is a system model which specifies the relations between all elements involved in an activity. In the following, the components of the ASM are specified. Additionally, the theoretical foundation of the ASM in the earlier discussed system models of Leontiev and Engeström is shown.

4.2.1.1 Elements of the Activity System Model

The ASM models an activity as a relation between a subject and an object, just like activity-theory (AT). Nevertheless, the system contains additional elements which share systemic relations with the subject and the object: different mediators and a context element, unknown in AT. Based on these elements the ASM can be used as an analytical framework which will be discussed in the next sections of this chapter. For now, the elements are presented:

- **Subject:** The subject is the actor in the system and stands for an individual or a group.
- **Object and Outcome:** The system realizes a transformation process from an object into an outcome.
- **Tools:** Tools capture productive mediators like material tools that produce a transformation of the environment and psychological tools that produce new information in the context of an activity based on a declarative system.
- **Rules:** Rules are declarative systems that constrain the activity system. At first hand, the difference between rules and psychological tools might not be obvious as both are declarative systems. The difference is the productiveness. Rules are possibly connected declarative systems that do not produce new information as they assure conformance of the activity to existing systems. Psychological tools are productive, as they are used in the context of an activity to produce new information. Dependent on the ASM, rules may be considered as tools. An example is the law that is a rule during everyday life. For the lawyer working on a lawsuit, the law is a tool that mediates an activity.
- **Workflow:** The workflow specifies the coordination of tool usage and object interaction.
- **Context:** The context element of the AT-SDM captures things that frame the activity without being a mediator, the subject or the goal of an activity. The use of the context element has specific benefits for the hierarchical decomposition of goals discussed later in this chapter (see section 4.2.2).

In Engeström's ASM mediators are only connected to entities (community, subject, goal). The explicit connection between the mediators in the ASM presented here stands for the regulative effects each mediator has on the other mediators. The idea is that the relation between the mediators shapes the whole mediation process. Rules influence tool choice and workflow. Tool choice influences rules and workflow. And finally, the workflow influences tools and rules.

All in all, the ASM of AT-SDM consists of subject and goal, mediated by rules, tools and workflow set in a context.

4.2.1.2 Generating the Activity System Model

The ASM just described is based on the activity systems introduced by Leontiev and Engeström. The creation of the ASM by following the principles used to create Leontiev's and Engeström's activity system are described in the following.

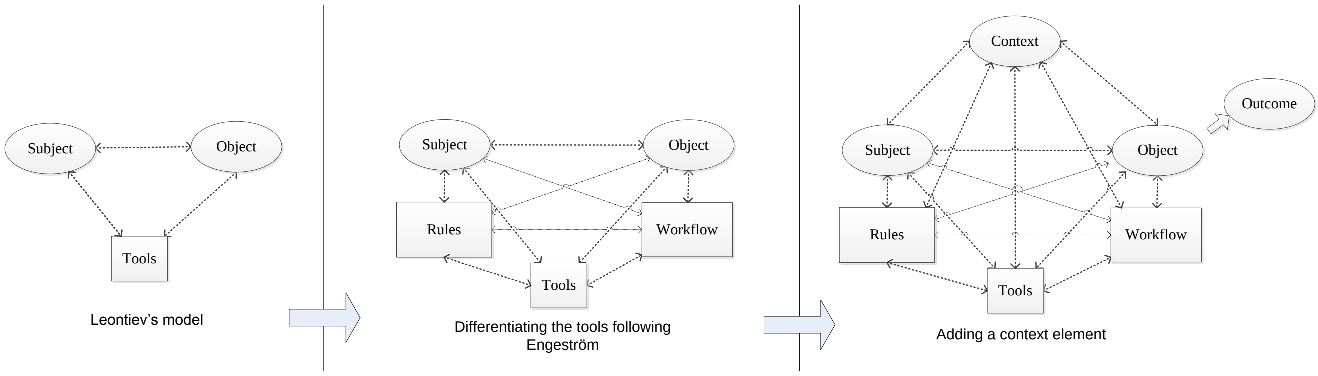

Figure 4.2.: The three steps of model construction.

1. **Lentiev's activity system foundation:** The ASM is based on Leontiev's acvitity system. Leontiev's ASM states that goal achievement of subjects is mediated by tools (see left side of Figure 4.2 and section 2.1.3.2).

2. **Mediator extension:** The first difference between Leontiev's system and the introduced system is the use of mediators. The ASM of the AT-SDM extends Leontiev's model to specify the mediator in more detail (see middle of Figure 4.2). Mediation in the ASM of the AT-SDM is closely aligned with the work on mediation by Engeström (see section 2.1.3). The mediator (tools) in Leontiev's ASM is not specified further. Vygotsky distinguished different types of mediating tools. Namely psychological and material tools (see section 2.1.1.1) which are classes that include a broad range of mediators. Engeström's ASM included two mediators to address social and cultural aspects of human activities: rules and division of labor. The ASM of AT-SDM uses three mediators: rule, tool and workflow. The choice results in a system model of three connected mediating systems: 1) subject, community mediated by rules 2) community, goal mediated by division of labor and 3) subject, goal mediated by tool (which is Leontiev's ASM).

   The ASM of the AT-SDM follows Engeström's conception, as rules and tools are used as mediators as well. For the use of rules and tools the same arguments hold that are used by Engeström (for details, see section 2.1.3.2). The third mediator of the ASM of the AT-SDM is the workflow. Using workflow as a mediator is unique for the ASM of the AT-SDM compared to other ASM's. The workflow is related to the division of labor used by Engeström. The difference between workflow and division of labor is an important characteristic: time. Division of labor does not consider time. In fact, ASM's tend to be agnostic to time as orchestrating factor. Using the workflow as a mediator embeds time as an orchestrator of concurrent processes in the mediation process of the ASM. The benefits of the temporal dimensions will become obvious when the heterarchy of multiple ASMs and the coordination of the systems based on activation is discussed (see section 4.2.2).

3. **Context element and object:** The goal is decomposed into an object and an outcome. The system structures a process of transforming the object into the outcome (e.g., text documents about research are processed into a state-of-the-art document). Furthermore, a context element is introduced (see right side of Figure 4.2). AT considers activities as contextualized [140, p. 34]. Nevertheless, none of the mentioned ASM's make context explicit. The relevance of the context object unfolds in the next section which introduces the heterachical relation between ASMs.

4.2.2 Activity System Heterarchy

ASMs can be related to each other just like an activity can be related to an action. This embeds each ASM in a structure of subordinate and superordinate systems and models activity as a set of more or less related ASM's (see Figure 4.3). This process closely follows the heterarchy design applied within action regulation theory (ART). The difference is that the heterarchy does not connect goals but that it connects activity systems. The resulting structure specifies relations between cognitive processes and interactions with the real world. Within the structure systems can have effects on each other. An ASM which addresses a complex goal is realized based on a set of subordinate ASMs which themselves could trigger subordinate ASMs which stand for the interaction with the real world. A heterarchy is used which means that the superordinate ASMs affect subordinate ASMs but also vice versa. This is relevant, because a perception on a low level of the heterarchy may modify the higher level ASMs. As an example, a subject might plan to write an email (ASM with the goal of mail writing) but clicking on the icon for the email (ASM some levels below the the mail writing ASM) program shows an error message. The perception of the error is propagated up to the mail writing ASM which is reorganized accordingly.

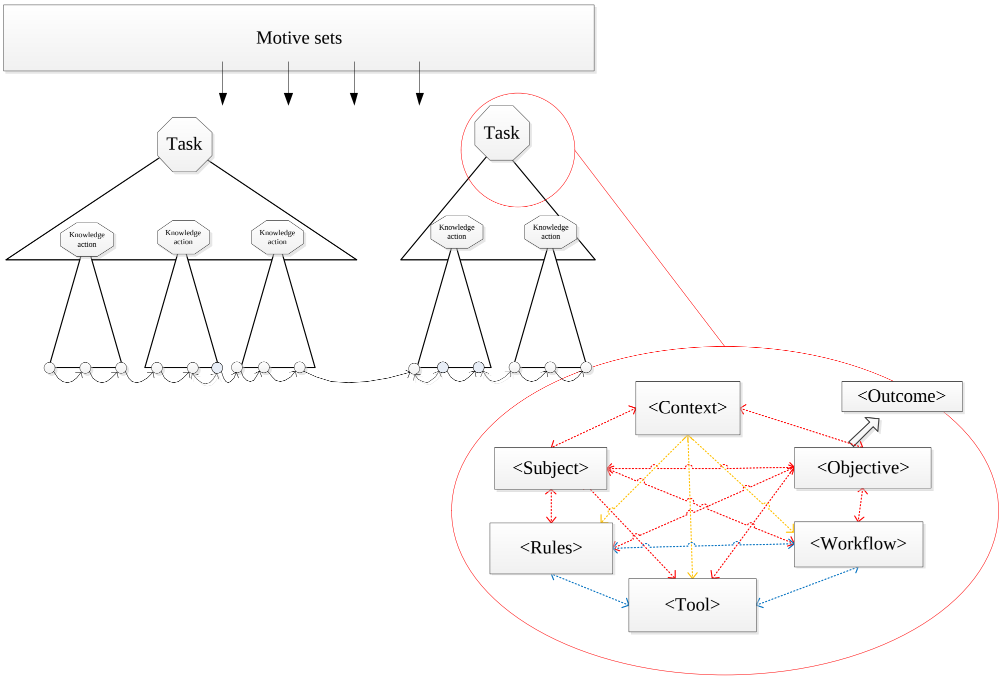

Figure 4.3.: The ASM heterarchy with the motive layer on top.

In the following, the highest layer of the heterarchy is described. Finally the role of the context element within the heterarchy is presented.

4.2.2.1 Motives in the Heterarchy

Superordinate to the heterarchy are guiding motives that have no ASM. Guiding motives structure strategic decisions and generate high level ASMs (a similar usage of motive as described in AT, see section 2.1.2.2). On the guiding motive layer, different motives may exist that are contradictory, e.g., working quickly and working with best possible results. The subject prefers certain motives when an initial ASM with an object is created. The ASM heterarchy emerges due to the subject's motive set. The subjective perspective helps to analyze processes of resource allocation of a subject and to identify relations and side effects of different systems existing in parallel.

4.2.2.2 Role of the Context Element in the Heterarchy

The heterarchy uses the context element as an inter-model mediator, i.e., elements that belong to the superordinate ASM can become context elements which constrain an activity on a lower level of the heterarchy. The heterarchy decomposes activities into activities with a higher specialization. An ASM is decomposed into a set of subordinate ASMs. The decomposition of activities results in a specialization of the mediators, subjects and the identified subgoals. E.g., When a word processing software is a tool on a higher level of authoring demand, the lower levels will directly address specific features of the word processing software as a tool. The word processing software degrades from a tool to a context factor for its features for the ASMs subordinate to the text production ASM. This process of specialization modifies the role of an element within an activity system. E.g., the work processing tool loses its mediator status, a group of subjects is decomposed into a single subject, etc. Thus the context element collects those ASM elements that lose their status of being mediator, subject or goal due to the specialization of the system.

The context captures elements of different granularity which degrade from specific roles (subject, mediator, object) to framing factors in subordinate systems due to system specialization. This relation mediates between different levels of the heterarchy.

4.2.3 Intermediate Results

The AT-SDM is a method to realize a context of use analysis and a requirement specification which considers the cognitive processes involved in work execution. The method creates a model of the domain of interest (context of use) and provides means to analyze this (requirement specification). This section has introduced the basic elements of the model, the ASM heterarchy. Cognitive and interaction activities are connected in a heterarchy of ASMs. Each ASM specifies an activity as a system model for a mediated process of goal attainment in a given context. The model is based on the principles of AT and ART.

The following sections will introduce additional aspects of the model and of AT-SDM as a whole to enable method application to investigate information work in the next chapter.

4.3 Activity System Properties

The goal of this section is to define activity system properties. The properties are required to consider the coordination between different activities within the heterarchy.

- **Activation:** An ASM is active if the subject actively pursues its goal (see section 4.3.1).
- **Balance:** An ASM or the heterarchy is balanced if all elements interact within the goal attainment without any obstruction (see section 4.3.2).
- **Complexity:** Complexity denotes the effort required to execute an ASM(see section 4.3.3).
- **Distance:** The distance is a function to specify the effort required to switch between two ASMs (see section 4.3.4).
- **Awareness:** Awareness denotes a subject's consciousness of an activity. A low awareness decreases the activation likelihood of an ASM (see section 4.3.5).

Based on the properties it is possible to model a subjects's decisions for activity switches. This is a basic requirement to use the AT-SDM for an analysis of information work.

4.3.1 Activity System Activation

An ASM has the property active which can be true or false.

- **Definition:** An ASM is active during the time, the subject actively pursues the goal of the ASM. A goal is pursued actively by conscious cognitive treatment or by physical interaction.
- **Description:** Activating an ASM means that the subject of the model spends cognitive and physical resources at the respective point in time working on the active activity systems. An active ASM produces effects. On the lowest level it generates interaction with the environment. On higher levels, it maintains the subordinate levels. Generally, more than one ASM can be active at the same time. However, the amount of active activity systems for a single individual is limited by cognitive and physical capacity.

  No direct relation needs to exist between the activated ASMs. In fact, the activation of an ASM does not necessarily activate the superordinate or all subordinate ASMs. This intuition is guided by the ideas of threaded cognition. Threaded cognition argues that the hierarchy of goals is not always conscious.[1]
- **Hint:** Activation uses the concept of time. The use of time is justified by the workflow element which provides the required temporal notion in the ASM. The workflow element allows to consider an activation as successive to another activation.

[1] Asking a person at a desk what he is doing shows the limited activation. The person will answer on the activity but continuing to ask "And why are you doing this" reconstructs the hierarchy of ASMs. The individual will be able to construct the hierarchy but now and again the individual has to think to reactivate the superordinate ASM.

4.3.2 Activity System Balance

An ASM has the property balance which can be true or false. If the balance is false, one can also say that the system is in tension.

- **Definition:** A system is balanced if two conditions are met. 1) The elements of the system do not interfere negatively with each other. 2) The system does not interfere negatively with another system. Negative interference refers to effects that hinder or deny the transformation process from object to outcome within the ASM.

- **Description:** Activity systems model the process of pursuing a goal based on the interplay of the elements of the system model. The elements influence each other. The tools are used in a workflow according to the rules within a given context to transform an object into the outcome. However, the interplay between the elements might be complicated (e.g., a rules might deny the use of a tool or a tool is not applicable within a predefined workflow). If the unobstructed interplay of the elements is not given, there is a negative interference. Similarly, two ASMs which depend on each other due to their integration in the same heterarchy may interfere negatively with each other (details on this will be provided later in this chapter, see section 4.4.2).

  The consequence of negative interference is a tension within or between systems (the term tension originates from existing work on tensions 1) between ASMs [149] 2) between the elements within one ASM [85]). If a tension exists in an ASM, the system is considered as unbalanced. If the relation between two or more systems contains tensions, the set of systems is unbalanced.

  To avoid misunderstanding, two aspects of balance need to be highlighted. First, an unbalanced system is not necessarily unproductive. The lack of balance only complicates the productiveness of the system. Second, considering a system as balanced does not mean that the system is the optimal system to achieve a goal (e.g., with respect to time, complexity...). The balanced system is only the most productive system with the elements involved, e.g., the activity system to create a book copy produces a perfectly balanced ASM for a writer in a scriptorium copying line by line by hand. The use of a copier is a balanced ASM which has a much higher productivity.

  The heterarchy is balanced if all ASMs in it are balanced. Each ASM on any level in the heterarchy may lose its balance due to unexpected effects during the execution of the respective activity.

- **Hint:** A loss of balance triggers regulation based on mediator modification. Mediator modification refers to the process of replacing the element contained in a mediator (rules, workflow or tools) by another element which is accessible within the context of the ASM (e.g., the tool WordPad is replaced by Microsoft Word). The replacement operation has effects on the whole ASM and can be used to resolve the tension. However, the element which becomes a mediator in the replacement process may cause new tensions. Therefore, the replacement process must be closely observed to avoid such unwanted effects.

  Changing a specific mediator might be very complex or even impossible. An ASM models the activity of a subject which is not necessarily completely shaped by the subject. The subject needs to follow existing constraints (e.g., the workflow might be predefined, rules like laws can be predefined). Therefore, the stability of a mediator needs to be identified before a replacement is performed.

  If regulation fails to reestablish balance in the ASM, the instability is forwarded to the superordinate system in the heterarchy.[2] Once, balance is regained, all ASMs subordinate to the first stable level are modified or deprecated.

4.3.3 Activity System Complexity

Complexity is a property of an ASM.

- **Definition:** The complexity denotes the effort required to attain the goal of the ASM. Complexity is specified by a value on a scale (e.g., numbers or classes like “low”, “medium”, high).

- **Description:** The complexity of an ASM refers to the complexity of transforming the object into the outcome. The complexity can be specified for a ASM based on the involved elements.

- **Hint:** Complexity is only specified on an ASM as a whole. Even if an element of the ASM also has a complexity specified (e.g., the goal), this does not necessarily result in a high ASM complexity. Subjective capabilities, work techniques and higher mental functions (see section 2.1.1.1) determine the method set accessible by the subject to solve a goal. The more work techniques are on an operation level, the less exhaustive is the actual execution of work. For the goal example: a complex goal might be realized with an ASM which has a low complexity because the ASM includes an elaborate tool. On the other hand, if a tool is missing, the simple goal of opening a can has a high complexity.

[2] The process is similar to the discussion with respect to failure in section 3.1.1.

4.3.4 Activity System Distance

The distance is a function to specify the effort required to switch between to ASMs.

- **Definition:** Distance measures the effort required to switch between two activity systems with a switch as deactivating one ASM and activating the other ASM. Distance is a function with two ASMs as input and a distance value on a scale as output. The scale for distance can be implemented differently (e.g., numbers or classes like "low", "medium", high). The distance from an ASM to itself is always defined as the lowest possible value on the scale.

- **Description:** The switch between two activities requires effort. For the deactivated activity the subject needs to memorize the latest status information and needs to clean the workspace (i.e., for information work closing applications and saving files). For the activated information, the required facts need to be recalled and the workspace needs to be prepared accordingly (i.e., for information work starting applications and accessing information objects). The effort required depends on the distance of the ASMs. If they deal with a similar topic and if they require a similar workspace, the switch is simple. If the topic is very different and the modification of the workspace is complex, the distance is high. The description shows that the term changeover time used in industrial assembly is related to the distance term used for ASMs.

- **Hint:** For this dissertation a basic understanding of distance is sufficient. To provide this, the following list illustrates factors which influence the distance. However, for specific applications it will be necessary to describe the characteristics in detail. The distance of two ASMs can be described in terms of the similarity of the included elements. Distance depends on the similarity of the subject, the context, the mediators and the goals of the models.
  - **Subjects:** If both subjects are different, there is no similarity. If one subject is a subset of the other, then a partial similarity is given. Complete similarity is given if the same subject executes both ASMs.
  - **Context:** The context is an enumeration of elements that frame an ASM. The similarity of the context is an assessment of how complex a change from one context to another context would be. The specification of distance for some complex factors like location is simple. A change of a context *city Paris* to a context *island Juist in the North Sea* is obviously complex. For other context factors like two different persons in the contexts of two ASMs, it is no simple task to decide about the complexity. Rich, system-specific background knowledge can be used to determine whether two persons have a high or a low distance. Therefore, context similarity is an expert assessment that applies rich domain knowledge.
  - **Rule mediator:** Similarity is given if the rules are similar. Partial similarity is given if some part rules are similar or if one rule is a subset or a logic product of another rule. Similarity decreases further if no logic connection between the rules is given. No similarity is given if the rules of the two models are contradictory.
  - **Tool mediator:** Similarity is given if the tools are similar. Partial similarity is given if the tools of the two models belong to the same class of tools and if similar techniques can be applied to operate them. No similarity is given if the tools belong to different classes and require different interaction techniques.
  - **Workflow mediator:** Similarity is given if the workflows are the same. Partial similarity is given if one workflow includes the other workflow. Similarity decreases based on the occurrence of similar work steps in the workflows. If no two work steps in the workflows are similar, then no similarity is given.
  - **Goal:** Two different types of goal similarity exist. One type is goal similarity which is given if both activity systems strive to achieve the same goal (e.g., build the same building, win the same election). Abstract similarity is given if two goals belong to the same goal class (e.g., build a building, win an election). Partial similarity is given if a goal is a subclass of another goal. If the goals share no class relation, then no similarity is given. For a more detailed analysis of goal similarity, the similarity of the objects the activities are performed on and of the outcomes needs to be identified additionally.

4.3.5 Activity System Awareness

The awareness specifies a subject's consciousness of an ASM.

- **Definition:** Awareness is a property of an ASM and specifies a subject's consciousness of the respective ASM. The awareness translates directly into the activation likelihood of the respective ASM. A subject's awareness is a limited resource. ASMs exist which have no awareness at all. Awareness within a ASM can be modeled by values that sum to one.

- **Description:** The cognitive capabilities of a subject are limited resources. Attention is a mechanism used to allocate the resources in the most effective way. Awareness is a crucial element in the self organization. Only those ASMs a subject is aware of are considered by the subject and become activated based on self interruptions.

This section has specified properties for ASMs. The properties are required to analyze coordinative cognitive processes in work execution. The property activation specifies the active work on an activity. Only based on activation it is possible to specify activity switches and multitasking within the ASM heterarchy. The properties complexity and distance are introduced which denote the effort required to execute an activity (complexity) as well as the effort to switch between two activities (distance). With the awareness the likelihood for an activity to be activated due to an activity switch is specified. An example of the property application for an ASM heterarchy is provided in Figure 4.4.

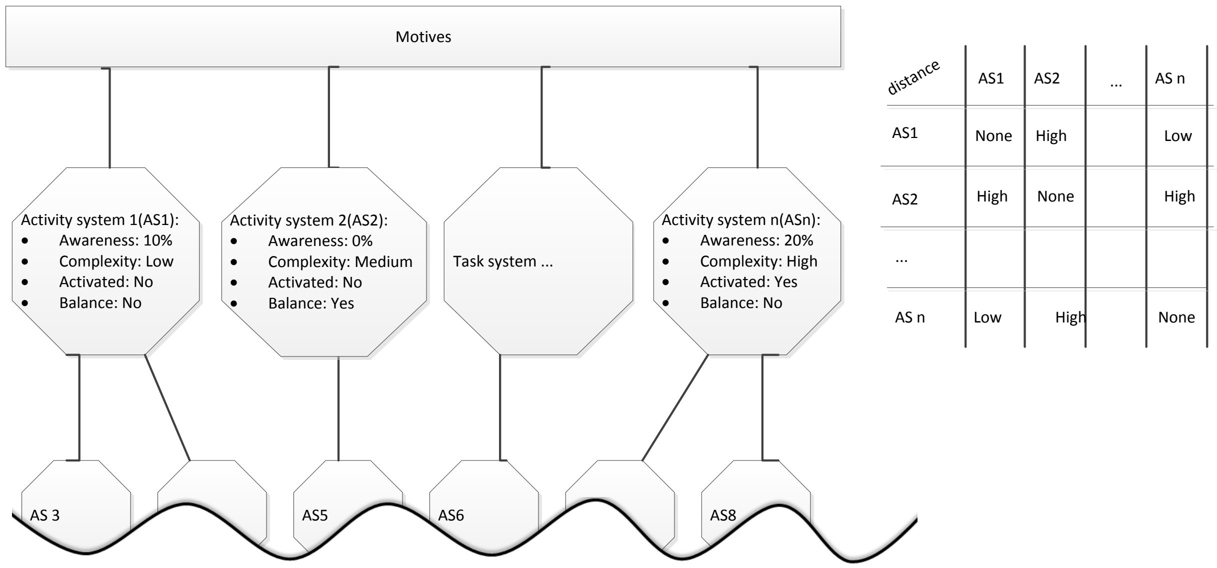

Figure 4.4.: Example application of the properties activation, complexity, distance, balance and awareness for an ASM heterarchy.

These properties are used in the following section for a tension analysis of the ASM heterarchy. The tension analysis provides rules to apply the properties to analyze the heterarchy.

4.4 Activity System Tension Analysis

ASMs are not only descriptive but they can be used to identify tensions. Tension identification in ASMs is a method to identify model element interactions within and between systems that decrease the productivity of systems or sets of systems. Examples for tension analysis in ASMs are the methods proposed by Mwanza [191] and Engeström [84].

Two main tension classes can be distinguished: intra-model tensions and inter-model tensions. Intra-model tensions occur within one ASM and address the interaction between the subject, object, mediators and the context (see section 4.4.1). For the identification of the intra-model tensions an analysis method is described. Inter-model tensions occur between different ASMs in the heterarchy (see section 4.4.2). The tension analysis uses the properties defined in the previous section. For the identification of inter-model tensions a set of tension patterns is specified. The intra-model tension analysis as well as the tension patterns are the result of a set of workshops on the analysis of tensions conducted between the authors of [76]. Based on discussions and example systems for work situations of a group of people as well as for individuals tension sources and regularities of tensions have been identified and are presented in this section.

4.4.1 Intra-model Tension Analysis

The first type of tensions to be specified is the inter-model tension. Inter-model tensions emerge within an ASM due to a negative interference between the included elements. To identify the respective tensions, a three step approach is proposed (see Figure 4.5). Each step analyzes a subset of the activity system elements with respect to their interference on the process of transforming the object into the outcome within the activity.

1. **Mediator Triangle:** First, the mediators are analyzed to figure out their interference within the activity. Therefore, a triangle of all mediators is analyzed. Each mediator is focused and analyzed with respect to its relation to the other mediators. The following three questions are answered in this process: Are the used tools and the workflow conforming to the rules? Are the tools useful with the workflow and the given rules? Is the workflow aligned with the tools and the rules?

2. **Subject-Mediator-Context Triangles:** Second, the mediation between subject and context is analyzed. The suitability of the mediators for the subject in the given context is analyzed. Three triangles are analyzed: Subject-Context-Rule, Subject-Context-Tool, Subject-Context-Workflow. The question is always if the subject can use the mediator in the given context.

3. **Mediation Squares:** Third, the application of the mediators in the context of goal achievement is analyzed. The actual realization of the goal is analyzed in the final step. The squares Subject-Context-(Object/Outcome)-Rule, Subject-Context-(Object/Outcome)-Tool and Subject-Context-(Object/Outcome)-Workflow are analyzed with respect to the question if the mediators in the given context allow the transformation of the object to the outcome.

Based on the three described processes the tensions within one ASM are unveiled.

4.4.2 Inter-model Tension Patterns

An inter-model tension can occur between two or more ASMs that belong to the same heterarchy.[3] The tensions decrease the productivity of at least one of the ASMs involved in the tension.

In the following, patterns for inter-model tensions are defined. To identify the inter-model tensions, the ASMs within a heterarchy needs to be analyzed with respect to the characteristics of the specified patterns.

The inter-model tension patterns defined in the following belong to three different groups: activation organization, model compatibility and parent-child relation. The inter-model tension patterns provided here focus on heterarchies which involve the same subject for each ASM. Respective patterns exist for groups as well but are not reported here.

4.4.2.1 Activation Organization

In the following the organization of ASM activation is discussed as source of inter-model tensions:

- **Simultaneous Activation:** Overlapping activation refers to the activation of multiple ASMs at the same time.

  Description: More than one ASM with the same, single subject is active within the ASM heterarchy. Two limiting factors for the simultaneous activation of more than one ASM exist. First the subject's cognitive capabilities limit the number of ASMs because each active ASM requires cognitive capabilities (e.g., working memory and short term memory). Second, the subject's physical features limit the number of parallel interactions (e.g., a hand can only perform one motoric action at a time). If two active ASMs exceed the cognitive capabilities or require the simultaneous use of the same physical feature, a tension occurs. Therefore, tensions depend on the complexity and distance of the simultaneously active ASMs.

- **Switching Activation:** Switching activation refers to an activity switch. An active ASM is deactivated in favor of another ASM which is activated.

  Description: The subject switches from one ASM to another ASM. The switch involves the deactivation of an active ASM (memorizing the status of the system) and the activation of another ASM. This activity switch involves the cognitive and physical processes that have been described for the interruption (cf. section 3.2).

  The switching is a source of tensions if the ASMs involved in the switching process have a high distance or have a high complexity (these characteristics have been identified in section 3.2.4).

[3] To realize the requirement of the same heterarchy for systems with different subjects, a superordinate node is required that comprises the union of both subjects which is separated on lower levels. This node can also be the motive node.

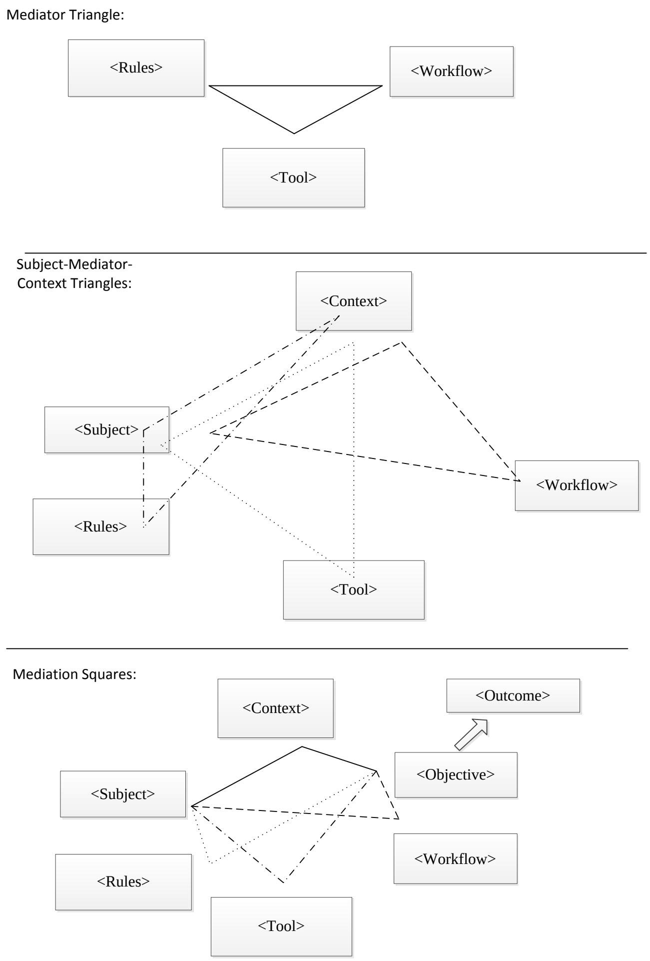

Figure 4.5.: The three analysis steps to identify intra-model tensions.

- **Limited Awareness:** The tension occurs if one or more ASMs have a very low or no awareness at all.

  Description: The awareness of the subject is a limited resource which is distributed among the ASMs. Therefore, the awareness of certain ASMs decreases if the overall number of ASMs exceeds a subject's specific threshold value. As a result, the subject is prone to forget the respective activity and will not continue unfinished work. The respective ASM is abandoned until a stimulus increases the awareness.

4.4.2.2 Model Compatibility

The second group of inter-model tensions to be defined considers ASM compatibility. Two ASMs are not compatible if the execution of one system requires a complex or infeasible regulation of the other system.

- **Goal Incompatibility:** Goals are incompatibility if a logical contradiction exists between two goals.

  Description: Goal incompatibility is a tension which makes the completion of at least one of the involved ASMs impossible. ASMs are incompatible if their goals are contradictory: For a single subject, the realization of one goal is not possible without making the realization of the other goal impossible. The contradiction is not necessarily conscious for the subject. If the individual becomes aware of the contradiction, it needs to modify one of the goals. If the subject does not become aware of

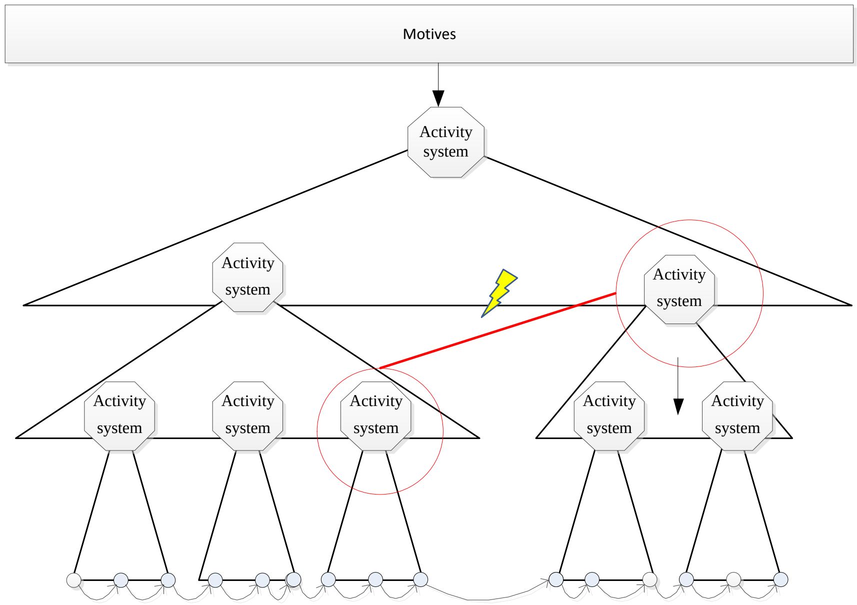

Figure 4.6.: Inter-model tensions.

the contradiction, the coexistence of the contradictory goals continues although the completion of both goals is impossible. It is not possible to address the tension by regulation activities because the outcome needs to be modified which results in a different activity (cf. the relevance of the goal for an activity in section 2.1.1).

- **Model Modification and Invalidation:** Two models may influence each other as they operate on the same objects. One model modifies or transforms objects that are expected to be in a certain state in the other model. As a result a tension occurs which complicates the execution of at least one of the involved ASMs. In an extreme case one model even invalidates the other one. This invalidation is given if one ASM consumes or destroys the object required in another ASM. This tension pattern can be addressed by regulation activities because the activities address the object.

  At first glance, incoherence and invalidation seem closely related. If one goal contradicts another goal, one might assume model invalidation. Still, both are separated to address that subjects are not perfectly logic beings. Considering western society as a subject, numerous goal incoherencies can be generated, e.g., with respect to ecological or economic goals. Goal incoherence only leads to model invalidation if a logic consistency of all ASMs is required[4].

  Description: The modification, consumption or destruction of mediators or context elements in one ASM will modify the structure of the other ASM if the ASMs share the object. Although the subject is the same, the connection of both is not necessarily known by the subject.

4.4.2.3 Parent–Child Relation

A third class addresses inter-model tensions which only exist between ASMs that are connected in a parent-child relation.

- **Inappropriateness:** An ASM can only be inappropriate with respect to its superordinate ASM. An ASM has the purpose to support the realization of the goal or motive of the superordinate ASM. Thus a superordinate ASM has an expectation towards the outcome of a subordinate ASM. An inappropriate ASM does not support the realization of the superordinate goal or motive. The outcome is intended to serve as object in the superordinate system. Therefore, the tension pattern emerges between the object of the superordinate system and the outcome of the subordinate system.

  Description: If parent and child ASM are both executed by the same subject, the inappropriateness will become obvious when both systems are active. The subject recognizes that an expected outcome is not produced by the subordinate ASM

[4] An example from science fiction literature/film for goal incoherence that leads to model invalidation is the computer HAL in 2001: Space Odyssey. HAL has two incoherent orders which leads to the invalidation of the respective activity models and the assassination of the crew.

(e.g., an information worker decides to create an illustration to describe a complex passage within a document. During the work process on the document and the illustration the information worker recognizes that the illustration will not help to understand the text and, therefore, he modifies the illustration). To address the tension, the information worker can change the subordinate ASM or the superordinate expectations if possible.

- **Separatist Tendency:** The separatist tendency is a tension between a superordinate ASM and a respective subordinate ASM. *Description:* A subject creates an ASM to address a requirement of a superordinate ASM. The created subordinate ASM has a high complexity and disconnects from the original superordinate ASM. The subject executes the goal even if the original superordinate system might be modified or is deprecated and the original cause is not given anymore (e.g., a subject starts to research facts about a new technology to be integrated in a software. Due to the complexity of the topic the subject needs to review much information and to build different prototypes which consumes more and more time. As an effect the awareness of the ASM is high. Even if the superordinate ASM is modified to use a different kind of software the subject might continue the study of the new technology.).

4.4.3 Intermediate Results

Methods for the identification of intra- and inter-model tensions have been specified. While a method to identify intra-model tensions has been provided, the identification of inter-model tensions relies on an analysis of inter-model pattern membership.

4.5 Context of Use and Requirement Specification

This section specifies the methods realized by the AT-SDM, the context of use analysis and the requirement specification. The previous sections of this chapter have described the model AT-SDM is performed on. The heterarchy of ASMs is a model to specify and analyze activities which involve cognitive decision processes. In the following, the creation of a context of use based on this model is described (see section 4.5.1). Based on the tension analysis problems within the context of use become obvious. Such problems are addressed in a model transformation process which is used to identify requirements (see section 4.5.2).

4.5.1 Context of Use

A context of use basically is an ASM heterarchy with the properties described in this chapter. To create a context of use for a domain of interest, the respective activities need to be identified and their structure and relations need to be used to build the respective ASM heterarchy. To leverage the benefits of ASM, the considered domain should involve dynamic ad-hoc work processes. Only for such work processes which involve cognitive processes of work coordination the AT-SDM provides additional knowledge about the domain.

Initially, a state of knowledge about a domain of interest needs to be generated based on techniques like ethnographic studies. This state of knowledge is the foundation for the creation of the ASM heterarchy:

1. **System specification:** To create the different ASMs the domain knowledge needs to be structured in terms of goal directed activities. For each activity involved elements need to be classified according to the system elements (subject, context, rule, tool, workflow, and goal).

2. **Heterarchy construction:** To create the heterarchy the relations between goals need to be identified.

3. **Specify situations:** The identified ASM heterarchy provides an overall understanding of work execution in the domain of interest. However, to analyze the work more closely the heterarchy needs to be designed to specify relevant work situations. For those situations of interest specific heterarchies can be created to investigate further into the characteristics of the situations.

4. **Tension analysis:** Once a heterarchy of ASMs is constructed (for arbitrary but fixed tasks or for a set of specific situations), the model can be analyzed, using the discussed methods of intra- and inter-model tension identification (see sections 4.4.2 and 4.4.2). The result of the process is the identification of a tension set.

Based on the described process a context of use is specified which can be used for the requirement specification.

4.5.2 Requirement Specification

Requirements identified by the AT-SDM result from a structured process of model transformation. The ASM heterarchy is transformed to a balanced state. Based on the modifications required to reach this balanced state requirements are elicited. In the following, the respective process of requirement specification is described in detail.

The context of use includes a set of tensions which complicate the execution of activities and thus complicate goal attainment. The goal of the system design is to create a system which addresses existing tensions to facilitate goal attainment. Therefore, those tensions which need to be addressed by the design solution have to be selected.

Once, tensions to be addressed by the design solution have been identified the ASMs involved in the tensions are transformed. The goal is to transform each ASM involved in a tension to gain a balanced state with respect to the selected tensions (cf. the description of modifications in section 4.3.2). The transformation can be informed by a state of the art review. This can help to avoid problems by considering the experience gained with existing solutions.

Each element of each ASM as well as each complete ASM can be subject to modifications. However, the openness is limited. Each modification may produce side-effects. First, resolving a tension may create other tensions. Second, certain elements cannot be changed for a ASM and do not allow transformation or modification (e.g., predefined work processes are generally not open to changes). Therefore, it is important to create the balanced ASM heterarchy in a systematic and controlled manner. In the following one process to identify requirements is described.

1. **Parent ASM selection:** The root of the modification process is chosen. The desired state will only address inter-model tensions that exist for the root and its children. Inter-model tensions are only addressed if at least one of the ASMs involved in an inter-model tension is the root ASM or an ASM subordinate to the root node.

2. **Perspective selection:** Modifications for a balanced ASM heterarchy are intended to be minimal. This restriction limits the side-effects on the ASMs. The restriction is realized by requiring the selection of a type of mediator which will be modified: tools, rules or workflow. If the modifications only address rule mediators, the resulting system design will focus on the adaptation or creation of rules. If only the tool mediator is modified, the system design will result in the (re-)design of a tool. If only the workflow mediator is modified the system design will generate a new structure of work execution. Thus, the decision for a modification can be seen as a *perspective selection* for the whole system design process AT-SDM is applied in.

3. **Parent node tension resolution:** Once the perspective is selected the parent ASM can be analyzed. The selected mediator needs to be modified appropriately to address tensions in the parent ASM. Once the modification concepts have been integrated into the ASM, a new inter- and intra-model tension analysis needs to be conducted. If new tensions occur that are not acceptable, the modification is reiterated. In this process effects of the modifications for the other ASM elements are identified and changes to regain a balanced system are performed. If no unacceptable tensions emerge anymore, the system is balanced with respect to the tensions to be addresses by the system design. The next process step to specify the modifications starts.

4. **Generate new ASMs:** Each modification within the ASM heterarchy stands for new activities which have been introduced. Therefore, new ASMs need to be created for each system modification. The new system specifies how the modification is achieved (e.g., an inter-model tension of lost overview between a *tool: word processor* and an *object: document collection* is addressed by a *tool: collection browser*. For the introduced *collection browser* it is necessary to specify a system model which captures the interaction with the browser). For each created subordinate ASM, an inter- and intra-model tension analysis is executed. The process continues until no unacceptable tensions emerge anymore.

5. **Integrate new subordinate ASMs with existing subordinate ASMs:** After the new ASMs have been created, different systems may exist which have a similar goal. This results from the creation of new ASMs based on the mediator modification. For two ASMs with the same goal two options exist: integration or replacement. If the model constructed for the actual state includes relevant information that is not addressed in the newly created ASM, then integration is performed. In the integration process the additional information is taken over into the new system. The old system is deleted and an analysis of intra- and inter-model tensions is executed. If no information needs to be transferred, a direct replacement occurs which means that the old ASM is deleted.

6. **Extract requirements:** The new subordinate ASMs contain functional and non functional requirements for the modified or created mediator. Each generated ASM stands for a requirement to be realized by the—dependent on the chosen perspective—tool, rule, or workflow to be developed in the system design process.

4.5.3 Intermediate Results

This section has specified the application of the ASM heterarchy for a context of use analysis and the requirement specification. The details about the elements of the ASM heterarchy, including the properties and the tension analysis are applied within the two described processes. Overall, the methods realize the AT-SDM which is the basic contribution of this chapter. The extension of the user-centered design (UCD) by the context of use provides a requirement specification which considers the cognitive processes involved in activity execution. Therefore, the UCD with AT-SDM can be used to analyze information work in the next chapter.

4.6 Summary

This chapter has provided the system design method which will be used to create software to address memory failures in information work.

The chosen system design method is UCD. The reason for this decision is the focus on the user perspective in the system design process. The first steps of UCD are a context of use analysis and a requirement specification. To execute UCD, methods need to be chosen to realize these steps. For an analysis of information work it is important to consider the cognitive processes involved in the coordination of the work process. However, no method was identified which considers the cognitive processes appropriately. To address this shortcoming, the AT-SDM was developed which addresses cognitive processes in the analysis.

The AT-SDM is a method which realizes the first two steps of the UCD system design method, namely the context of use analysis and the requirements specification. AT-SDM basically creates and analyzes ASM heterarchies. The specific advantage of this structure is the consideration of cognitive processes and the effects of those processes on the interaction with the world. Properties for ASM heterarchies which facilitate the design of decision processes among different activities have been specified. Based on the properties a tension analysis technique was presented which is used to identify those elements within the heterarchy which complicate goal attainment in terms of inter-model tensions and intra-model tensions. Processes for context of use analysis and for requirement elicitation for the ASM heterarchy have been specified. The processes include the creation of the heterarchy, its analysis and the systematic transformation to address relevant tensions and derive respective requirements.

To sum up, the chapter delivers the foundation for the creation of a software to address memory failures. The UCD with the AT-SDM is used in the remainder of this dissertation to create and validate a system design. The next section will specify the context of use and will deliver requirements which start the first iteration of the UCD method.

5 Requirements Engineering for Information Work at the Computer Workplace

The remainder of this dissertation uses user-centered design (UCD) with the activity theory based system design method (AT-SDM) (see section 4) to design concepts and methods for software to support mnemonic processes involved in information work at the computer workplace. Two iterations of the UCD system design method will be executed. As a result different concepts and methods are developed which have been implemented and evaluated in two prototype applications. This chapter describes the context of use analysis and the requirement specification of the first design iteration.

The AT-SDM is applied following the defined process (see section 4.5). The goal is to analyze information work in terms of activity system model (ASM) tensions and to derive requirements based on a systematic transformation of the tensions. The context of use is a ASM heterarchy which models the domain of interest in terms of ASMs. To create a context of use, the information work ideal type (cf. section 3) is translated into an ASM heterarchy. This translation helps to bring the loosely coupled features of interruption based coordination and work techniques into a systemic relation (see section 5.1). Based on the context of use, requirements are elicited in a three step process:

- **Tensions:** Intra- and inter-model tensions are identified with a focus on those tensions that increase the likelihood of memory failures (see section 5.2).
- **State of the art:** A state of the art analysis of existing information work support tools that address memory failures and simplify work organization is conducted to identify characteristics of existing solutions, respective benefits and problems (see section 5.3).
- **Requirement elicitation:** The work model is transformed to resolve tensions from a tool perspective. The transformation is informed by a state of the art analysis. Based on the transformation requirements for an information work support tool are derived (see section 5.4).

5.1 Context of Use I: Information Work at the Computer Workplace

The major challenge tackled in this section is the translation of the information work ideal type into a context of use in terms of an ASM heterarchy. The context of use analysis follows the process specified in the previous chapter (see section 4.5). The description of the process is subdivided in two sections. This section specifies the ASMs the information work heterarchy is composed of. The next section identifies inter- and intra-model tensions in the ASM heterarchy.

5.1.1 Information Work Ideal Type Translated to Activity Systems

The ideal type specifies information work execution with respect to the execution process that unfolds at the computer workplace. To build a context of use for the AT-SDM, the basic elements of the ideal type need to be encoded in the ASM heterarchy. The different facets of the ideal type are considered differently within the ASM heterarchy. In the following a first overview of the different means of encoding ideal type characteristics within the ASM heterarchy is given:

- **Basic characteristics:**
  - *Effectiveness:*
    - CHARACTERISTIC: Effectiveness highly depends on the autonomy of the subject. Based on delegated goals the subject identifies subgoals and specifies acceptance criteria for those subgoals. Those subgoals and acceptance criteria are dynamic and can change during the work process based on regulation. Thus effectiveness is in a state of permanent evolution. The perspective closely follows action regulation theory (ART) and uses goal heterarchies to model the regulative processes.
    - CONSIDERED BY: The ASM heterarchy extends the goal heterarchies used for the ideal type. Therefore, the regulation process is already covered within the ASM heterarchy structure. The ASM heterarchy requires more information about the activity related to the goal which needs to be specified within the ASMs.

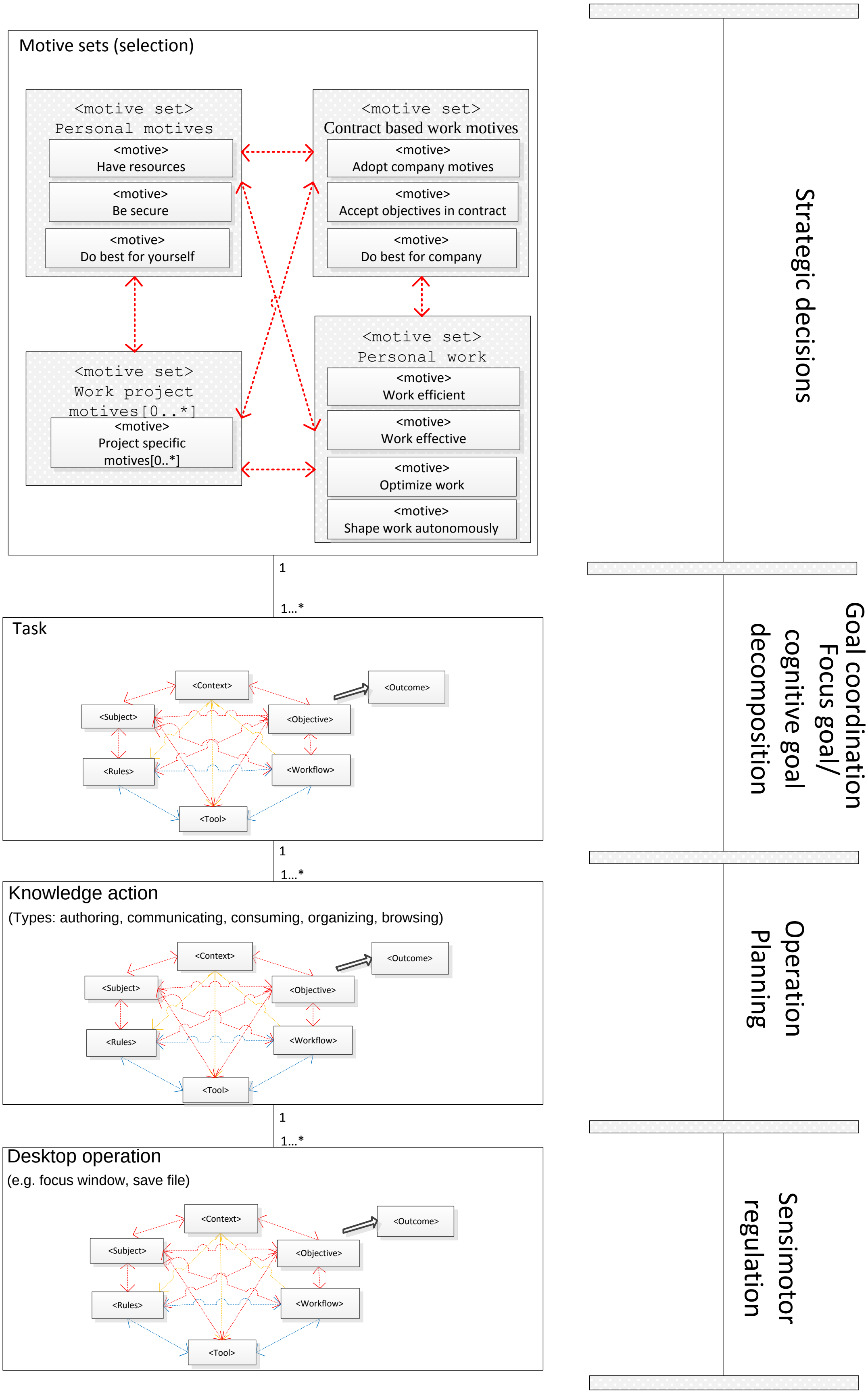

Figure 5.1.: Information work activity system heterarchy.

- *Efficiency:*
  - CHARACTERISTIC: A high efficiency requires a good operational cognitive image which covers a deep knowledge of the anticipated outcome, ways to fulfill the anticipation and adaptations to address unexpected events during the work execution.
  - CONSIDERED BY: The ASM heterarchy can be used to model the operational cognitive explicitly in terms of mental models. Mental models have a tool characteristic as they mediate the process of transforming an object into an outcome. At the same time the mental model is an object on its own. The model is transformed during task execution and contributes to the creation of the outcome (cf. the work on internalization in section 2.1.1.2).
- *Relevance of information:*
  - CHARACTERISTIC: The ideal type elaborates on the relevance of information for information work execution and characterizes the threat of information overload.
  - CONSIDERED BY: The role of information is specified within the ASMs. Information most frequently is used as object which is transformed into an outcome. Nevertheless, information can also occur as tool. Information overload will be specified further in a tension analysis.
- *Relevance of the information workplace:*
  - CHARACTERISTIC: The computer workplace is characterized by the computer, respective applications and the formalized interaction with the machine.
  - CONSIDERED BY: The computer as well as applications serve as tool within the ASM. The means of interaction influence the workflow element.

- *Coordination:*
  - CHARACTERISTIC: The work process of the information worker is largely coordinated based on interruptions.
  - CONSIDERED BY: The interruption based coordination is addressed by the properties specified for the ASMs. In this respect the properties distance and priority are of specific importance.

- *Technique:*
  - CHARACTERISTIC: Knowledge actions and desktop operations have been specified as basic units of work, information work processes are composed of.
  - CONSIDERED BY: Actions and operations are considered as activities within the ASM heterarchy. Action ASMs are addressed by subordinate desktop operation ASMs.

The translation of the characteristics is applied in the following. The information work heterarchy is specified in detail. Motives are addressed by tasks which in turn are addressed by knowledge actions. Desktop operations realize knowledge actions. The specified activities are arbitrary but constant.

5.1.2 Information Work Heterarchy: Motive Layer

Information workers have motives which generate goals. The goals generated based on motives are addressed by tasks. Motive characteristics are inherited by the task and influence the system elements (context, rules, tools, workflow and objective). This is of specific importance for contradictions. Motives may be contradictory. If a task is created based on contradictory motives, the contradictions are propagated into the activity systems as tensions which increase the likelihood of unbalanced ASMs.

Despite the fact that many motives of an information worker are subjective and will highly differ even for homogeneous groups of people, some motives can be considered relevant for most information workers (considering the analysis of work in general and of information work in particular, cf. chapter 2 and 3). On the one hand, fundamental needs like vital security interests are addressed by personal motives. On the other hand, information worker specific motives are personal work motives (e.g., expert culture), contract based work motives and project specific work motives. Tensions between the different motive types are likely and some motives will be ignored in favor of others. However, ongoing ignorance of certain motives most likely results in a feeling of dissatisfaction experienced by the subject [74].

The following list elaborates on the different motive types considered relevant for information work and discusses likely tensions:

- **Personal motives:** Personal motives address basic needs of human beings, like the need to have resources to continue living, to be secure or to take care of oneself.

- **Personal work motives:** Personal work motives follow the personal understanding of work execution the information worker has established. They reflect an expert culture and generate specific needs like being efficient, being effective, optimizing personal work or a predilection of autonomous work.

- **Contract based work motives:** Subjects declare by contract that they work for an organization. With the contract, the subjects adopt company motives and accept objectives defined in the contract. The contract is closely related to the personal motives of success and safety and is intended to underpin the personal work motives which may unfold and gain more strength by executing the contract based work. Next to the positive aspects, the contract based motives contain limitations of the personal work motives and the personal motives. An example is the commitment of the subject to certain regulations and rules which are likely to be conflicting with an idea of autonomy expressed in the personal work motives.

- **Project based work motives:** The work project motives stand for motive sets which result from the commitment to projects for a certain period of time. Project motives can be in conflict with contract based work motives, as the requirements of a project might be in conflict with existing work rules (e.g., the role of a project lead might be in conflict with the role of being an employee). Especially the project motives are an important source of tasks which then need to be aligned with other motives.

5.1.3 Information Work Heterarchy: Task Layer

The second layer of the information worker's activity system heterarchy addresses task activities. Information workers constantly work on a variety of different tasks. Each task can be represented by an ASM which stands for the activity to pursue a goal the information worker has committed to, based on motives. The coexistence of multiple tasks results in a large set of coexisting ASMs. The information worker needs to decide which task activity is executed. In terms of AT-SDM, the decision for a task activity means that an ASM is activated. The respective activity switch means deactivating one ASM in favor of activating another ASM.

Tasks can be decomposed into subtasks or are decomposed into knowledge actions. The elements of each task ASM are the following (see Figure 5.2):

- **Context:** Each task is set into an organizational, a spatial, a social and an environmental context. The organizational context is a formal frame defined by the organization the information worker works for, comprising the organizational structure with its hierarchy, its processes and its relation to the information worker. The spatial context refers to the place the task is performed in, e.g., the office. The social context involves people that are relevant in the context of the task, e.g., colleagues. The environmental context refers to other things (physical or cognitive) that frame the task without mediating it (e.g., other companies, society, culture etc.).

- **Objective and Outcome:** Tasks realize outcomes. While working on a task, the outcome is anticipated based on an operational cognitive image and produced based on a transformation of the object. Once the outcome is produced, the task goal is pursued. An outcome is an observable state of the universe. For information work this is a certain type of information or information effect (cf. section 3.1.3). To give two examples, persuading colleagues of an opinion can be an outcome as well as the creation of a document can be an outcome.

  The object of a task activity system is a complex and dynamic element:

  - *Information objects:* The object contains a plethora of information objects involved in task execution and which potentially change during the activity execution (cf. information overload in section 3.1.3). The same holds for the mental model which might have an enormous complexity and might be modified while the activity is executed.

  - *Mental model:* As a specific type of higher mental function, the mental model captures and organizes information about a specific domain. The model is transformed during task execution and contributes to the creation of the outcome. An important characteristic of the mental model is its double nature. Next to being an object, it is a tool as well. It is an object as it is one source of the produced outcome. It is a tool as it mediates the activity execution when it comes to the understanding of newly gained information. This explicitly is one of the core elements of activity-theory (AT): the production of the subject within the activity.

    An example is the creation of a text about combustion engines. The mental model mediates the interaction with the object. At the same time, the mental model is an object which is used to produce the text. The mental model results from years of formal education and years of work on combustion engines. While the information worker interacts with related information objects, the object nature modifies the mental model, thus modifying the mediator. A cyclic process of modified mediation and object transformation is part of the process that produces the outcome, the final text about combustion engines.

- **Tools:** Computers with applications to create, modify, access and disseminate information are the main mediators of information objects for the work considered (cf. relevance of the computer workplace in section 3.1.4). Additionally, the information worker applies higher mental functions which mediate information understanding. Higher mental functions might be preferred over the computer if they are simple to apply and are not error prone (e.g., multiplications are made in the mind up to a certain degree of complexity. More complex calculations are done with a calculator). A specific group of higher mental functions are mental models which structure the individual understanding of a specific domain. As discussed above, mental models are a mediating tool as well as an object for the activity systems.

- **Rules:** The execution of tasks is especially shaped by the rules given by the organization, e.g., the corporate culture and the rules the individual has created for himself, e.g., the expert culture. These rules generate constraints for the execution of the activity.

- **Workflow:** A mixture of process autonomy and process heteronomy generally shapes the information work workflow (this is a basic characteristic of the ideal type, see section 3.1). On the one hand, the information worker is an expert with the autonomy of deciding how to realize a task. On the other hand, the information worker follows predefined processes. Additional influence factors need to be considered like deadlines as temporal constraints.

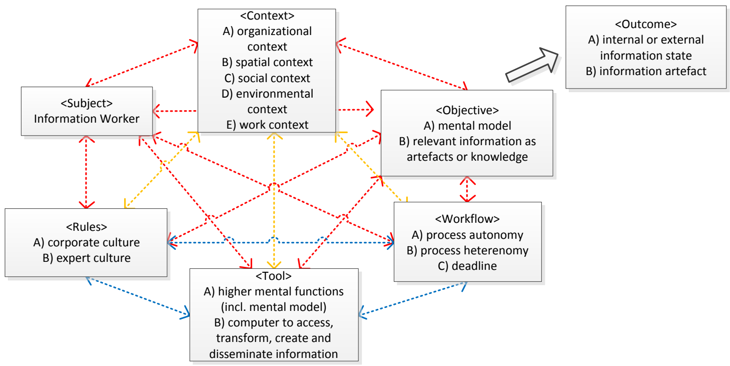

Figure 5.2.: Activity system for a generic task in information work.

5.1.4 Information Work Heterarchy: Knowledge Action Layer

Knowledge actions are work techniques which are combined to execute a work task in information work. Respective knowledge actions for work at the computer workplace have been identified (cf. section 3.3.3.2). Information workers apply knowledge actions to realize task execution processes. In the goal hierarchy of a subject, knowledge actions are on the level of operation planning (see section 2.2.4).

In the following, the activity systems for the different considered knowledge action types are discussed. The specification builds on the work techniques described in the information work ideal type (see section 3.3.3). Knowledge actions are individual work techniques. Therefore, autonomy is of high relevance and is used as a workflow element in each reported knowledge action system.

The context of a knowledge action is derived from the superordinate task ASM. For the generic task ASM used here, each knowledge action has a context, composed of the same elements. Therefore, they are reported separately (a visualization of the different knowledge actions is given in the appendix, see section A.2, Figure A.2).

- **Context:** The context element of a knowledge action inherits all context elements of the superordinate task elements. Therefore, the organizational, the spatial, the social and the environmental context exist as well. Additionally, different elements which are mediators on the task level become context on the knowledge action level:
  - *Tools:* Task tools become knowledge action context as the tools to execute knowledge actions are more specific (e.g., authoring is done with a word processor).
  - *Rules:* Cultural rules become context as they do note mediate knowledge actions.
  - *Workflow:* The deadline/relevance of the work is not directly considered on the level of knowledge actions which puts this workflow element of the task level into the knowledge action context.

Another similarity for all knowledge action systems is the carryover of the mental model with the domain knowledge from the task layer. The domain knowledge mental model retains its double nature of being an object and a tool for all knowledge actions.

- **Authoring:**
  - *Objective and Outcome:* The outcome of the authoring knowledge action is an information object (text, graphic, etc.) which contains content in a specific form. The object is a mental model which provides (as an objective) and applies (as a mediator) domain knowledge. The mental model develops during the work process and interacts with a selection of information objects and relevant knowledge. As a result the content of the authored document is created.
  - *Tools:* A content producer (e.g., word or graphic processor) mediates the authoring process. Based on the specific requirements of an authoring knowledge action, the information worker selects an appropriate authoring tool. Next to the mental model with domain knowledge, discussed for the task, a mental model of authoring exists which mediates the authoring process.
  - *Rules:* The subject follows many rules which are related to the production of information representations. These include language rules, style rules, domain rules as well as logic rules. The language rules cover grammatical rules of the semiotic system used. The style rules address the positioning of content and layout elements. Domain rules focus on the type of language or content appropriate for the specific domain. An example is that a mathematical text follows different rules than a letter to a friend. A logic rule refers to the need that authored information needs to be structured in a logical, concise manner.
  - *Workflow:* The subject knows different ways of authoring. For the activity system those authoring methods are chosen that help creating an intended type of content. As the knowledge action is an individual technique, process autonomy applies for the authoring knowledge action and all other knowledge actions.
- **Browsing:**
  - *Objective and Outcome:* The outcome of browsing is access to information of interest. This can be one or more information objects as well as one or more locations within an information object. The object of browsing is a mental model of the required information and a selection of information stores.
  - *Tools:* The information worker uses information access tools as well as tools for searching and browsing to realize the browsing knowledge action. A browsing specific mental model mediates the browsing process, including knowledge of media specific aspects like hyperlinks.
  - *Rules:* Browsing is limited by access restrictions. Restrictions may address information which is private, has specific copyrights or has a high security level.
  - *Workflow:* The knowledge action is realized by different trained browsing techniques. Examples of such techniques are associative browsing which browses around a selection of categories and modifies the categories frequently based on newly gained information or specific search which browses for a very specific piece of information. Process autonomy applies.
- **Communication:**
  - *Objective and Outcome:* The outcome of communication is the dissemination of information to a specific person or group.
  - *Tools:* Dissemination is supported by different tools which realize different dissemination techniques (e.g., unknown vs. known audience, synchronous vs. asynchronous dissemination, intended reaction vs. no intended reaction). The double nature of higher mental functions applies as described for the authoring. A mental model of the recipients in relation to the information mediates communication. In practice this mental model will include anticipated reactions or assumed triggered actions based on the dissemination of the information. The information which needs to be disseminated can be an information object as well as knowledge of the information worker.
  - *Rules:* Communication needs to reflect social rules which depend of the relation between the information worker and the addressee.
  - *Workflow:* The communication process highly depends on the tool selection (e.g., unknown vs. known audience, synchronous vs. asynchronous dissemination, intended reaction vs. no intended reaction). In all cases the process itself retains the autonomy of the information worker.
- **Organizing:**
  - *Objective and Outcome:* The outcome of organization is the arrangement of elements following rules. The object is composed of the things to be organized which are rearranged in the organization process.

- *Tools:* Organization functionalities are generally embedded in software tools or the operating system which mediate the organization process. A mental model of the organization rules or scheme additionally mediates the organization process.
- *Rules:* Organization schemes might exist which structure the data. Therefore, the compatibility of a newly applied organization scheme to the existing ones needs to be considered.
- *Workflow:* The mediator of the organization workflow is a trained techniques to realize the structure scheme selected as objective. The information worker applies the most appropriate technique to the activity. As for all other knowledge actions, process autonomy applies for organizing.

- **Consuming:**
  - *Objective and Outcome:* The outcome of consuming is the extension of memory based on consumed information. The object is composed of the information objects to be consumed.
  - *Tools:* The information worker uses tools for information visualization which mediate the consumption process. Additionally, the consumption process is mediated by a mental model of consumption processes.
  - *Rules:* The consumption process is governed by decoding rules which for texts are especially syntactic rules.
  - *Workflow:* The subject has a set of consumption strategies which are selected based on the type of information to be consumed. The consumption knowledge action retains process autonomy.

Earlier, frequent switches between different knowledge actions in a task execution process have been identified (see section 3.3.2). In terms of the model this means that different ASMs on the knowledge action layer are active in parallel (or are subject to frequent switches). This is interesting in the context of the tension pattern *simultaneous activation* (see section 4.4.2). The number of activity systems active in parallel is limited by the cognitive capabilities of the subject. Only if the ASMs have a low distance the activation of many different systems is feasible.

5.1.5 Information Work Heterarchy: Desktop Operation Layer

Knowledge actions are composed of desktop operations (cf. section 3.3.3.1). Desktop operations bridge the gap between cognitive processes and actual interaction with the environment. The authoring of a document is composed of text typed into a document, frequently saving it and accessing style functions by clicking menu items. Browsing is composed of clicks on hyperlinks, typing text into form fields and scrolling or zooming content visualizations. Activity systems for desktop operations may produce complex connected movements, e.g., focusing a certain window means the production of a set of coordinated body movements to move the mouse which moves the cursors over the upper bar of a window, clicking the mouse to trigger a window drag and moving the mouse again to move the window representation, coordinated by the perceived window movement.

Each desktop operation involves deeply internalized coordination of the body in conjunction with perceived modifications and internalized expectations towards the effects of an interaction which is guided by the trained rules of application design and the desktop metaphor. By this, the desktop operation stands for types of human computer interactions which materialize cognitive anticipations in interactions. Consequently, desktop operations are situated on the sensimotor level of the goal hierarchy (see section 2.2.4).

A large set of desktop operations has been identified in this dissertation (see section 3.3.3). In the following, the structure of desktop operations is discussed based on one generic desktop operation ASM. This serves as an example for the different types of desktop operations. The elements context, rules, tools, workflow and goal for the interaction ASM described are provided in Figure 5.3. The elements are explained in the following:

- **Context:** The context of a desktop operation includes the rules and the software tools of the enclosing knowledge actions. These do not mediate the desktop operation but provide a context of its execution. The mental model and the information objects of the enclosing knowledge action also become part of the context. This transition stands for the role of the enclosing objective for the desktop operation: it frames the operation without being its objective. Additionally, desktop operations inherit the context of the enclosing knowledge actions which results in a huge and complex context.

- **Objective and Outcome:** The outcome of a desktop operation is the realization of an anticipated effect specific for the desktop operation in the given context (e.g., opening a file as interacting with a file representation in the context of a word processor has the anticipated effect of visualizing the file content). The outcome may include the modification of information beyond the mere modification of a visual presentation. The object to realize this outcome is the state of the body of the information worker in relation to a perceived state of the device the interaction will take place with and in relation to a perceived state of the information the desktop operation focuses on.

- **Tools:** The body, the input device and the state visualizations of a focused representation are the tools to realize the interaction. The application focused by the enclosing knowledge action is not the mediating tool anymore. Therefore, the application is part of the context. The visualization of desktop action specific states mediates the interaction. The body, the input device and the state visualization may have different proximities. A touchpad which modifies the state visualization based on a body gesture has a high proximity to the involved tools. The use of a mouse to modify a text has a high distance as the mouse movement needs to be connected to the movement of the cursor on the screen. The distance has influence on the experience of the desktop operation mediation: the mapping of the effect of the body movement on the input device visualized in an application shows different complexities.

- **Rules:** Desktop operation rules stand for the social interpretation of body movements interpreted as interactions and gestures. A desktop operation produces gestures, changes the position of the body and shows the relation between the information worker and the interaction tool. If other subjects perceive the interaction, the gestures and social rules like distance to others need to be considered in the desktop operation performance.

- **Workflow:** The workflow is the process of coordinated body movement based on the perceived relations of the body, the device and the focused information (the objective), to realize the outcome.

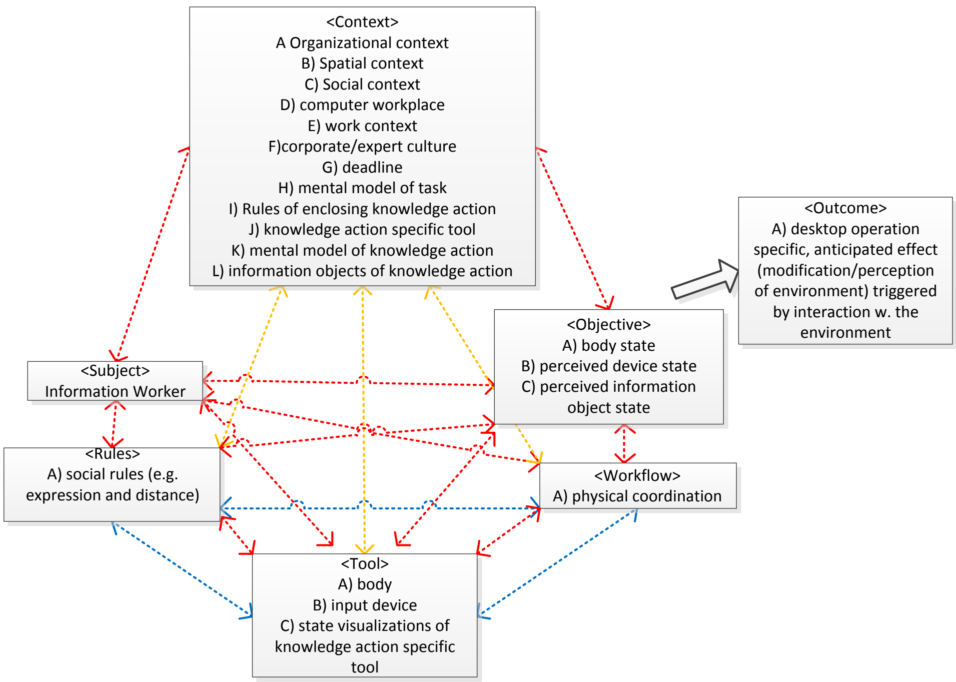

Figure 5.3.: Activity system for a generic desktop operation.

5.1.6 Intermediate Results

The information work heterarchy consists of four layers: motives, tasks, knowledge actions and desktop operations (cf. chapter 3). These layers have been specified based on the information work ideal type. This structure specifies the relations between those elements involved in the execution of activities. These relationships are analyzed in the following section.

5.2 Context of Use II: Tension Analysis

This section completes the context of use analysis with a tension analysis of the information work heterarchy. Tensions occur as intra-model tensions between system elements (see section 4.4.1) and as inter-model tensions between ASMs. Tensions between elements or systems indicate complicated activity execution. The goal is to identify those tensions which result in memory threats.

Once those tensions are identified the requirement specification process of the AT-SDM can be applied to identify requirements for software to address those tensions and to decrease the likelihood of memory failures.

The context of use process of the AT-SDM recommends the creation of ASM heterarchies for work situations of specific interest (see section 4.5.1). In the following, three work situations are considered more closely which derive directly from the information work ideal type:

- **Multitasking (see section 5.2.1):** A snapshot of a multitasking subject during the execution of information work.
- **Underspecified Work Process (see section 5.2.2):** A system which investigates into the way subjects address the uncertainty of the information work process.
- **Task Execution Activity (see section 5.2.3):** The task ASM is investigated to identify inherent tensions.
- **Interruptions (see section 5.2.4):** A snapshot of an interruption and the respective reactions.

For each work situation a dedicated ASM heterarchy of arbitrary but constant activities is created. The analysis focuses on the task level and considers related knowledge actions with respective desktop operations only as parts of the task systems. For an analysis of information work support, the task level is an obvious choice as the tasks shape the actual implementation of the respective knowledge actions and desktop operations. A tension analysis on the knowledge action or desktop operation level would result in specific support demands of knowledge actions and desktop operations which is out of scope for this dissertation.

The following report does not include all identified tensions. First, those tensions which have no obvious relation to mnemonic processes are excluded. Second, tensions between mediators of high stability unlikely to be modified are excluded as well[1].

5.2.1 Tensions I: Multitasking

The first situation is a snapshot of information work execution as conveyed by the ideal type: an information worker has to work on many different tasks. Some of those tasks are active in the moment of the snapshot while other tasks are inactive.

The ASM heterarchy for this situation is provided in Figure 5.4. Motives generate many different task activities. Each task activity is decomposed in dedicated knowledge actions. The distance between the knowledge actions which belong to the same task can be considered as "low"[2] while the distance between the ASMs which belong to different task activities can be considered as "medium" or "high". The ASMs have different complexity degrees, ranging from "low" to "high."[3]

A set of tasks within the heterarchy is active, i.e., they are processed by the subject. Other tasks are inactive which stand for those tasks which have not yet been completed. The awareness of the subject is distributed among the tasks.[4] Due to the large amount of tasks, some tasks have a low awareness of the subject.

Each ASM has a high complexity because it contains a large number of information objects and contain a large set of different mediators. Most ASMs contain complex mental models which stand for operational cognitive images which coordinate the activity execution.

5.2.1.1 Forget task

The first identified tension follows the limited awareness pattern. The consciousness of the existing task ASMs is modeled based on the awareness. Due to the limited cognitive capabilities it is infeasible for the subject to be aware of all tasks. As a result it is likely that tasks are forgotten.

- **Tension 1 – Forget planned tasks:** The subject fails to remember tasks and subordinate ASMs.
  - *Class:* Inter-model tension, Pattern: System maintenance problem
  - *Type:* Prospective memory failure
  - *Description:* The prospective memory based coordination among the numerous different systems is likely to fail resulting in missed optimal work situations, missed deadlines or—in the worst case—forgotten activities. The prospective memory based coordination presents itself as an inter-model tension based on the system maintenance pattern.
  - *Example:* An information worker has to work on a large set of different tasks with different deadlines. Without increased effort to keep up an awareness of the deadlines, the information worker is likely to forget a task or a specific deadline.

[1] Mediators which are defined by the organization or society are considered as very stable, e.g., social rules or predefined processes. An example is an intra-model tension on the task level between subject-context-workflow-object. The subject strives to produce an outcome based on an object with the workflow. The workflow includes predefined processes of the company which might be inappropriate for the outcome. As it is not likely that the predefined processes can be modified, this tension and similar tensions are not reported.

[2] For distance a simple classification set is used in the following, composed of "low", "medium" and "high".

[3] For complexity the same simple classification scheme is used which has already been introduced for the distance.

[4] To address the distribution percent values are used.

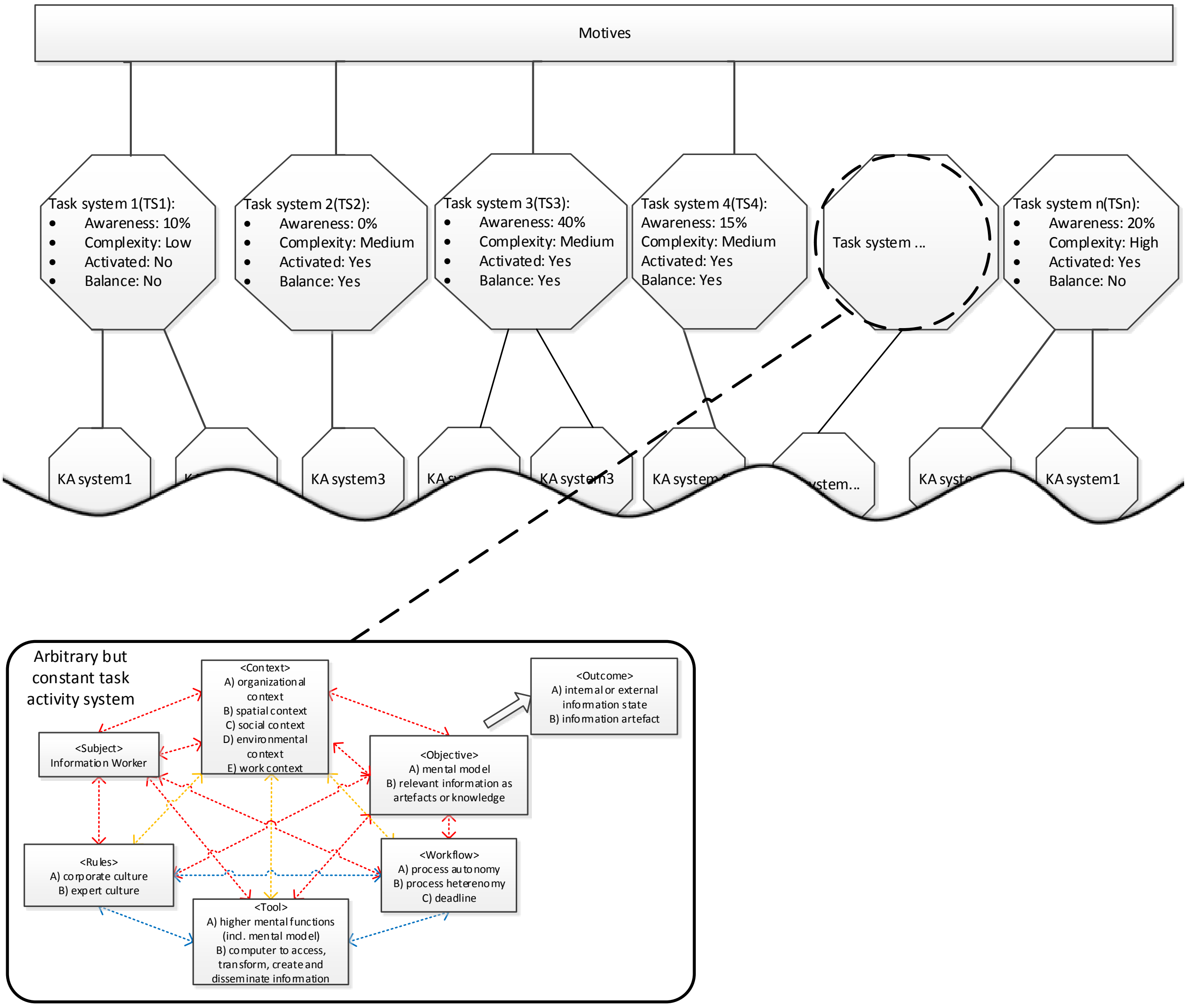

Figure 5.4.: Multitasking: Example of several activity systems which are active in parallel.

5.2.1.2 Lost overview

The ASMs within the heterarchy include a many information objects and many different mediators. If more than one ASM is active, the subject needs to keep an overview of the assignment of the used elements to different activities. The complexity of this activity is a tension.

The tension is an inter-model tension based on overlapping activation of the active activity systems. Each active system produces a physical representation of its execution, especially running applications with accessed information objects. Next to the tensions of resource allocation (e.g., parallel execution of computation intensive tasks), a tension mainly related to the working memory and the retrospective memory occurs. Interestingly enough, if the systems have a low distance which suggests the parallel execution, the memory threat increases. The difficulty of accessing elements for one activity system is already captured in tension 3. For more systems active in parallel, the threat of failing to access an object or to relate an object that is part of the physical working environment to the correct activity increases.

- **Tension 4 – Maintain multiple active activity systems:** The subject works on more than one activity system in parallel and fails to relate active elements to their ASMs.
  - *Class:* Inter-model tension, Pattern: Overlapping activation
  - *Type:* Retrospective memory failure
  - *Description:* Elements which belong to different activity systems make up the work environment of the subject. The subject needs to recall which element belongs to which system.
  - *Example:* An information worker creates two Excel spreadsheets to report performance data for different sales regions. For this, a business application is opened and the required data is copied from respective transactions into spreadsheets. As both systems share many elements they have a low distance and the parallel execution is obvious. Nevertheless, copying the correct information into the correct spreadsheet turns out to be a complex challenge. Unless the activity execution is operationalized, the subject frequently will need to think “did the information I just copied in this work sheet actually belong here, or does it belong to the other sheet?”

5.2.2 Tensions II: Underspecified Work Process

Most information work processes are underspecified. As a result, the subject needs to identify the optimal way of working on a task (see Figure 5.5, Nr.1). The subject will try different strategies to address the goal. Such strategies result in new ASMs which investigate into a solution strategy in terms of a knowledge action (see Figure 5.5, Nr.2).

The knowledge action might have a higher complexity than initially expected. Therefore, more and more awareness is given to the knowledge action, even more awareness than given to the task which initially triggered the knowledge action. A possible effect is the disconnection of the knowledge action from its origin task (see Figure 5.5, Nr.3). This is considered in the following tension:

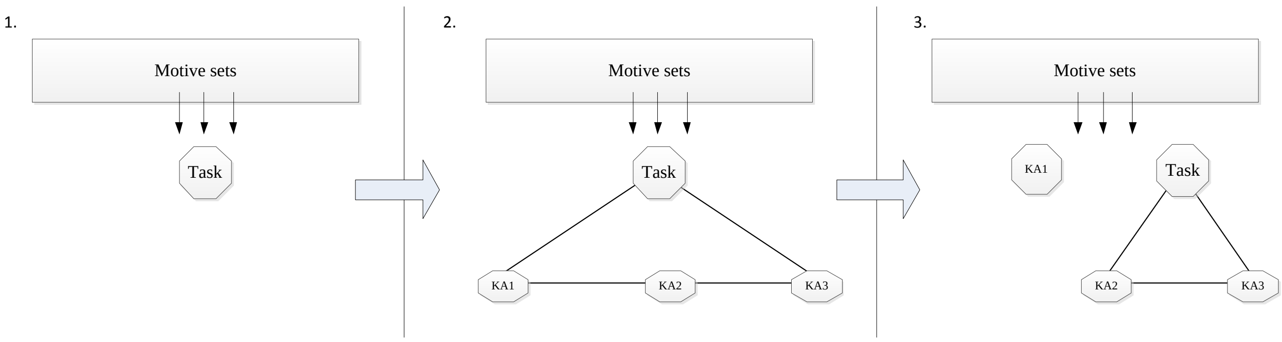

Figure 5.5.: The separation of a knowledge action from its original generating system.

- **Tension 5 – Separating subtasks:** A knowledge action separates from its origin and becomes a task.
  - *Class:* Inter-model tension, Pattern: Separatist tendency
  - *Type:* Retrospective and prospective memory failure

– *Description:* A very complex knowledge action may consume much time and the awareness increases. The subject begins to consider it as a task on its own without considering its root cause anymore. The outcome of the separated knowledge action might not be appropriate for the original generating task anymore.

  The separation of the knowledge action results in a lack of awareness of the work process and is a combination of prospective and retrospective memory failures. A prospective memory failure for the original parent activity as the subject forgets the requirements of the activity. A retrospective memory failure for the separated activity as the subject fails to recall the original cause of the activity.

– *Example:* An example is a consuming knowledge action which tackles a topic the information worker is not familiar with. The information worker begins to learn the topic which separates the knowledge action from its origin and makes it a task. The continued learning of the topic does not necessarily reflect which information was required for the task it originated from.

5.2.3 Tensions III: Task Related Tensions

The task ASMs are a central element of the information work heterarchy (see Figure 5.6). Therefore, they have been investigated closely by an intra-model tension analysis. The process of the intra-model tension analysis is provided in section 4.4.1.

The following tensions have been identified for the task ASM:

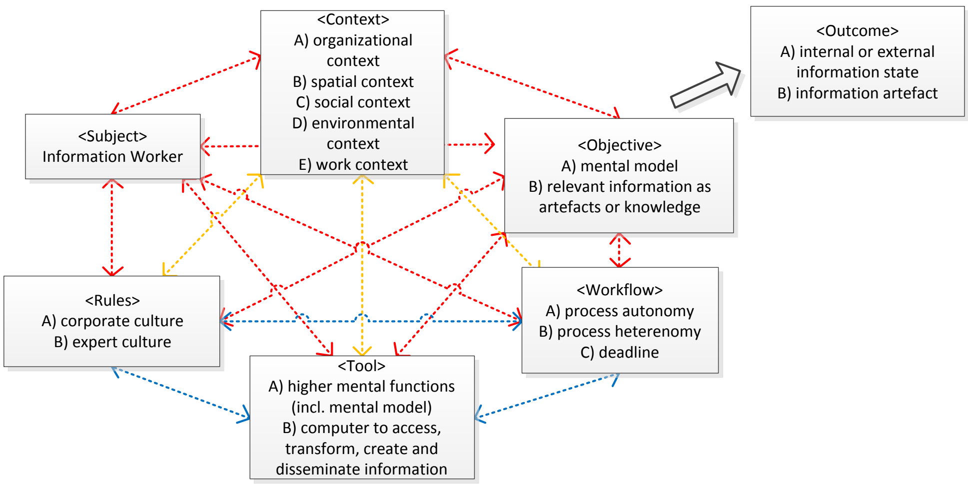

Figure 5.6.: Task activity system.

- **Tension 2 – Forget task status:** The subject fails to remember the status of a postponed task execution.
  – *Class:* Intra-model tension, Involved elements: Subject-Context-(Object/Outcome)-Tool-Workflow-Rules
  – *Type:* Retrospective memory failure
  – *Description:* Although an activity is recalled, the subject is unable to recall the activity's status, i.e., the required mediators and involved elements are known but the status of the transformation process from object to outcome is not remembered. This characterizes a failed recall of the operational cognitive image and can be considered as an intra-model tension which occurs between all elements of the activity system.
  – *Example:* Although an information worker remembers an unfinished task, the follow up is complex, as the operational cognitive image is not recalled. The subject is only able to derive the task status step by step, based on the interaction with elements which belong to the task.
- **Tension 3 – Maintain active activity system:** The subject fails to access the mediators and objects of the active activity directly
  – *Class:* Intra-model tension, Involved elements: Subject-Context-(Object/Outcome)-Tool

– *Type:* Retrospective memory failure

– *Description:* Accessing the mediators and objects required to work on an activity may be difficult. As activity systems frequently comprise a very large amount of mediators and objects, the execution environment will not provide simple access to all elements. Therefore, the subject frequently needs to access additional elements which belong to the activity and which were accessed earlier but which have not been accessed when the activity was activated. Retrospective memory failures may complicate the recall of the element positions and result in a duplication of search efforts. The access problem is an intra-model tension which occurs between a subject and a tool that does not allow a quick access of an object in the given context.

5.2.4 Tensions IV: Interruptions

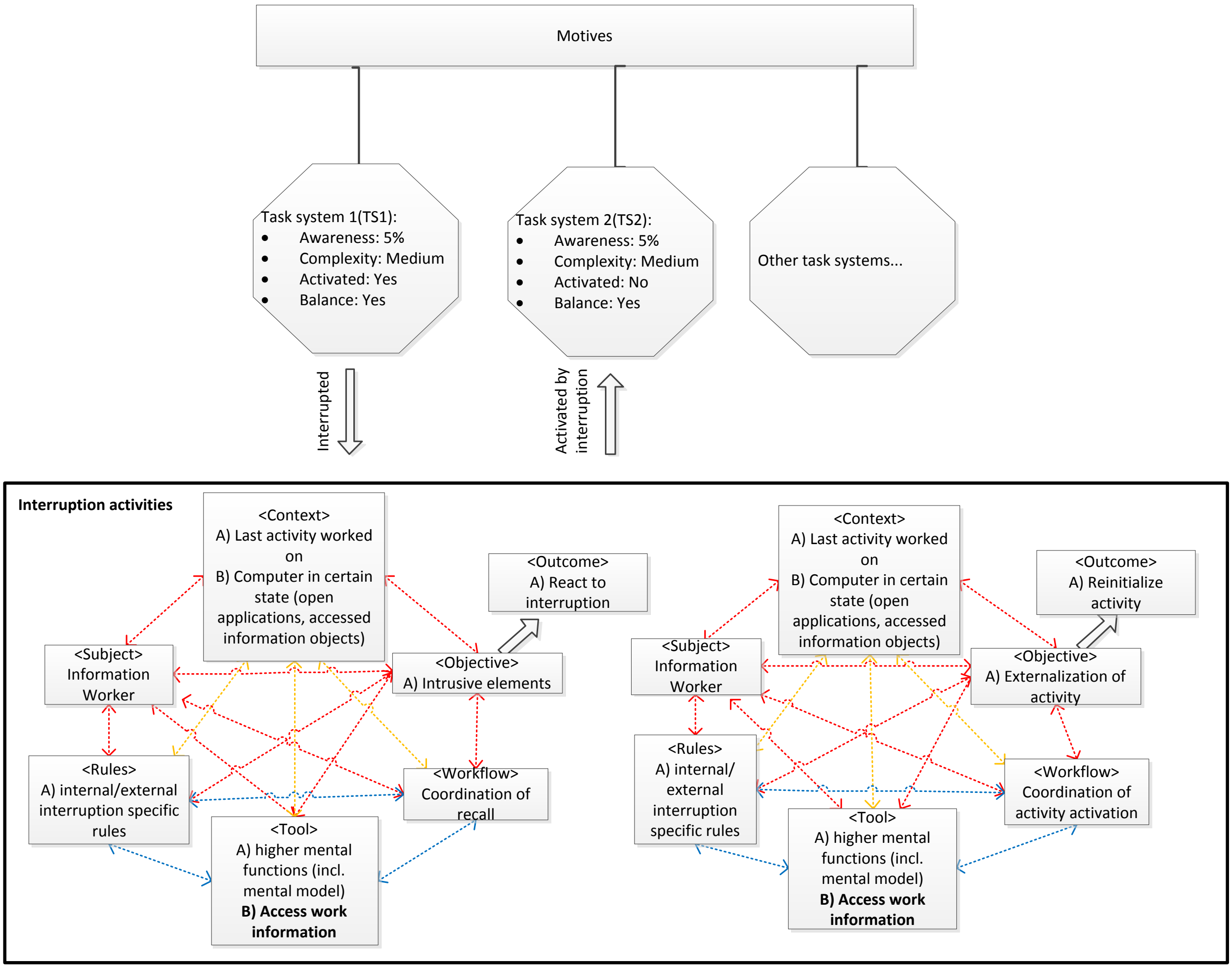

Figure 5.7.: Interruption and resulting interruption related activities.

The fourth situation investigated closer refers to the interruption based coordination of work specified in the ideal type. The execution processes of many different tasks are distributed over a period of time as they are frequently postponed and resumed, resulting in many switches between different ASMs with a potentially high distance. Each activity switch forces the subject to actively memorize a task related system state and to recall the details once a task is resumed (see Figure 5.7). The systems include a large amount of frequently changing mediators, objects and context factors which additionally complicate the mnemonic processes. Therefore, a tension of type switching activation has been identified:

- **Tension 6 – Interruption:** An interruption triggers the identification of an interruption target system and requires an activity switch.
  - *Class:* Inter-model tension, Pattern: Switching activation
  - *Type:* Prospective and retrospective memory failure
  - *Description:* Interruptions can be understood as stimuli which are incompatible with an active ASM and, therefore, destroy the balance of the active system. This description paraphrases the earlier given description of interruptions with switch decisions in terms of ASMs (cf. section 3.2.2). The individual is forced to exclude the stimuli forcefully from the conscious perception to regain balance. Alternatively, the stimuli are transferred to a new or existing and compatible ASM.

    The inter-model tension of switching activation occurs between an activated activity system and the activity system which becomes activated due to stimuli. If a new system or existing system is activated, a period of overlapping activation follows: the initial system which was interrupted and the system addressing the stimuli are active in parallel. If the maintenance complexity of both systems is high, one system will be deactivated. The complexity of interruption directly refers to the distance between the interrupted ASM and the interruption target ASM. The higher the distance, the more complex the identification and the activation of the system targeted by the interruption. Activation involves the cognitive focus of certain aspects of an activity as well as the access of objects and physical mediators in a specific context.

    Individual and external interruptions can be distinguished. Although the negative impact of internal interruptions is considered less harmful (see section 3.2.3), the structure of both interruption types is similar. Internal as well as external interruptions present themselves as switches between different ASMs. As interruptions occur between systems, they are considered as inter-model tensions which follow the switching activation pattern. The interruption tension encapsulates tension 1 and tension 2 and tackles prospective and retrospective memory.
  - *Example:* An example is an information worker writing a text. A colleague enters his office and asks for the status of a project. The information worker will try to integrate the project status into the active activity system which tackles the text production. If this integration is not successful (taken that the colleague can't be ignored), the information worker starts to think about the project, starts to recall it while the text production is not in focus anymore and reports the project status. Returning to the text production is complicated by the required mental switching.

5.2.5 Intermediate Results

This section has identified tensions which threaten the successful execution of information work and are closely related to memory failures. For the analysis four information work situations have been modeled based on the information work heterarchy which has been specified in the previous section.

The analysis of those situations resulted in the identification of six tensions which are likely to complicate information work execution (see table 5.1). Most of the identified tensions are related to retrospective memory failures: information workers forget the status of activities or the elements involved in activities. Nevertheless, some tensions also address prospective memory failures because subjects are prone to forget the activities they have planned.

An important characteristic of all identified tensions is the way they are most likely addressed: by additional knowledge actions. Most tensions can be addressed by additional search effort, performing checks on data or by externalizing information. The disadvantage of the additional activities is the related threat of efficiency: search efforts are duplicated, vast amounts of information are externalized on post-its, etc.

Next to the six identified tensions related to mnemonic processes a large set of other tensions has been identified, especially with respect to the identification of execution processes for underspecified tasks. The tensions that emerged show the applicability of the AT-SDM to gain a better understanding of phenomena of information work, like prospective and retrospective memory failures. Additionally, it is apparent that many tensions not considered in this dissertation are relevant.

This section concludes the specification of a context of use for information work execution. The identified tensions are relevant for the requirement specification process reported in the next two sections. First, the tensions are used to structure the state of the art report in the next section which prepares the requirement specification. The next but one section specifies requirements based on the identified tensions and informed by the state of the art review.

5.3 Requirement Specification I: State of the Art

The requirement specification process builds on the structured transformation of the context of use to dissolve identified tensions. This process has been described in the previous chapter (see section 4.5.2). The process description recommends to conduct a state

Number	Name	Description	Type
Tension 1	Forget tasks	The subject fails to remember tasks and subordinate ASMs	Inter-model (Pattern: System maintenance problem)
Tension 2	Forget task status	The subject fails to remember the status of a postponed task execution	Intra-Model (Subject-Context-Object)
Tension 3	Maintain active activity system	The subject fails to access the mediators and objects of the active activity	Intra-model (Subject-Context-Tool-Object)
Tension 4	Maintain multiple active activity systems	The subject works on more than one activity system in parallel and fails to relate active elements to their ASMs	Inter-model (Pattern: Overlapping activation)
Tension 5	Separating knowledge actions	A knowledge action separates from its origin and becomes a task	Inter-model (Pattern: Separatist tendency)
Tension 6	Interruption	An interruption triggers the identification of an interruption target system and requires an activity switch.	Inter-model (Pattern: Switching activation)

Table 5.1.: Information work tensions.

of the art review to prepare the transformation process. The review informs the transformation process and helps to avoid design issues of existing solutions.

This section conducts the state of the art review. The review focuses on tools that address mnemonic processes to support information work at the computer workplace. Information work support tools are investigated under consideration of the tensions related to memory failures identified in the previous section. The result is a state of the art review guided by a tension based understanding of information work execution and related memory failures.

The selection process of tools considered as state of the art takes different factors into account. First, tools to support the individual information worker are focused, excluding collaborative solutions. Second, the tools need to consider information work as weakly structured work execution and focus on the operations related to mnemonic processes: remember what needs to be done, remember what needs to be accessed, organize access to information. Approaches that consider information work as very structured, e.g., by modeling workflows of information work, are excluded (see also discussion in [83]). As a result, most reported work belongs to the domain of personal information management. Third, the selection is based on relevance with respect to publication quality (e.g., considering conference/journal ranking or publishing association) and the citation count.

The tools have been identified by a review of relevant human computer interaction conferences (including but not limited by UIST, CHI, IUI) and keywords or classification related to human-computer interaction, information work support and related keywords. Another requirement was end user focus of the solution which is given if the proposed solution has a user interface for end users (e.g., the Swish application focuses information work but only provides an expert user interface to analyze work processes [204]). For tools that have evolved over a longer period of time, reported in different publications, the focus is given to the latest reported state of the tool (e.g., TV-ACTA [23] preceded by [24]).

The report is structured based on the following categories:

- **Activity planning:** The planning of activities relates to the domain of task management. Objects that stand for planned or running activities are created and maintained (see section 5.3.1).
- **Activity awareness:** To improve the personal awareness of activities, the executed activities are visualized ex post (see section 5.3.2).
- **Activity specific information object access:** Access of information objects that are related to an activity is supported (see section 5.3.3).
- **Activity specific interruption:** Data forwarding to the user is filtered based on performed activities (see section 5.3.4).
- **Information work data model:** Access to information involved in information work is provided based on a specific data model (see section 5.3.5).

The information collected in the state of the art review is summarized in Table 5.2. This section partly follows the work published in [248, 245].

5.3.1 Activity Planning (Act-Pln)

The planning of activities relates to the domain of task management. Objects that stand for planned or running activities are created and maintained. Generally, the task objects contain information about deadlines and attached information objects to improve the access of task specific information objects.

- **Solutions:**
  - *Basic task management tools:* The most basic and frequently used approach is task management. Products for personal or group task management are the Outlook Task List, the Activities extension of Lotus Notes [133] or web tools like Remember The Milk [130].

    Different research prototypes build on the idea of a central task list and extend it with different functionalities. Tools like TV-ACTA [23] focus on the organization of tasks with attached information objects. Other tools extend the idea of task management by an idea of information reuse. Examples for this are TaskNavigator [124], Task Assistant [209] and Kasimir [243] which reuse information based on patterns or subtask proposals. Some task management systems create interaction histories that collect user activities (e.g., opened documents). UMEA [143] or Sphere Juggler [190] use those histories to support the maintenance of the task list.
  - *Automated task management:* The CAAD [217] system is specific, as it proposes selections of information objects that are automatically detected as activity representations. The clusters can be named which makes the tool a retrospective tool that allows planning for long term tasks that are continued in the future.
  - *Task-centric desktop extensions:* TaskTracer [77] provides a tool landscape that largely extends the computer desktop by task specific services. Based on user specified tasks, the system offers different functions. First, the system extends the Windows Explorer to explore resources related to tasks. Second, the system realizes a virtual desktop manager, as all information objects not related to a task can be closed. The system integrates task data in the windows start menu and a toolbar.

    Activity-centric desktop computing approaches emphasize the activity of task execution and related work environment organizations. Therefore, the desktop metaphor of the computer system is extended to activity-centric computing. An activity-centric desktop maintains relations between the work of the user and the task list. Rooms [117] is a window management system to organize windows that belong to different tasks. Therefore, a user's applications are organized based on a room metaphor. Each room provides window placements, application windows can be placed in. A window can be shared among different rooms. Robertson proposes a desktop with a work focus in the center. Elements moved to the periphery shrink without disappearing completely from sight [225]. The 3D Window Manager organizes tasks as images in a 3-dimensional space. Based on the spatial metaphor, the user can organize himself [224].

    Different extensions of the windows taskbar exist. Smith proposes Groupbar which structures activities in a taskbar [264]. A very similar approach is given with activity based computing that also organizes elements in a taskbar but with three distinctions: activities are persisted, can be transferred to other devices and avoid conflicts with the existing desktop metaphor [20]. The Co-Activity Manager [128] is another activity-centric approach which provides an activity based taskbar to organize activities and simplify activity switches. The Giornata system extends the Macintosh desktop by a program layer, making it activity-sensitive based on interactions between elements belonging to the same activity, an activity list and a contact list [287].
- **Required Data:** The task planning generally requires the manual creation and maintenance of task objects. Some tools support the task creation based on information from software sensors (Sphere Juggler [190] and UMEA [143]). The CAAD system does not require manual effort to create task representations as document collections [217]. Only the names of the tasks need to be maintained manually.
- **Addressed tensions:** Activity planning addresses many identified intra- and inter-model tensions:
  - *Forget tasks (T1):* The task objects externalize existing activity systems and help to remember them.
  - *Forget task status (T2):* Many task objects can be enriched by attaching information objects. These objects help the information worker to remember the status of the task.
  - *Separating knowledge actions (T5):* Some solutions offer the capability to create subtask relations. This helps to remember the context a knowledge action emerged in. The separation of a knowledge action from its origin becomes unlikely in this case.
  - *Maintain active system (T3):* The attachment of information objects to tasks helps the information worker to quickly access the relevant information objects.

- *Maintain multiple active activity systems (T4):* Those approaches which integrate task management deeper into the operating system (Periphery Shrink [225], Rooms [117], Co-Activity Manager [128], 3D Window Manager [224], Groupbar [264], Activity Based Computing [20], Giornata [287]) help the user to retain an overview while different tasks are executed in parallel.
- *Interruption (T6):* The overview of existing activities based on the task objects helps the information worker to identify the task an interruption is related to more quickly and to realize activity switches in less time.

Solution effects: The externalization of tasks addresses most identified tensions of information work. The analysis of the tools shows one important aspect: most systems require the manual creation of data. In terms of the ASM this can be considered as adding a maintenance task system to the information work heterarchy. The maintenance task needs to be executed very frequently to keep up a good quality of the task externalization and to maintain the attached information objects. In the worst case, the new task occurs as a frequently triggered self-interruption which does not support information work but complicates it.

The use of software sensors like for the UMEA system and Sphere Juggler is useful to limit the maintenance complexity without avoiding the interruption completely. The CAAD follows a different approach as it automatically mines collections of information objects. Assuming that externalization is mainly relevant for activities that require several working sessions, the approach to identify activities in interaction histories is an elegant way to maintain externalized activity data. As CAAD is driven by ex post information object usage, there are different limitations with respect to activity planning. CAAD does not allow the actual planning of activities that are not yet started and the limited capabilities of maintaining information like deadlines to the information object clusters limits its usefulness as full-fledged task management system.

5.3.2 Activity Awareness (Act-Awrns)

To improve the personal awareness of activities, the executed activities are visualized ex post.

- **Solutions:** Social Wakoopa [129] and Rescue Time [131] are commercial applications that visualize how long an information object was accessed to improve the process awareness of the information worker. A similar functionality is delivered by the Outlook Journal [30]. Such visualizations are also tested in the domain of technology enhanced learning to align the processes of learning and teaching based on the consumed information [79]. PersonalVibe (memory triggers for task tracking) [37] helps in recalling and reflecting on past work with a specific focus on writing status reports. The system offers an overview of the documents and applications a user worked on for different days and helps to answer the questions “What documents did I work on last week” and “How much time did I spend on each document?” The respective data is collected via user monitoring. Feldspar [53] is an interactive and incremental association based information retrieval system. The system collects visited websites, emails, file events, etc. (data indexed by the Google Desktop Search). The user is able to identify elements by incrementally specifying characteristics of the searched objects. By involving the page visit history, a retrospective element is added to the search approach. The described tools do not allow the grouping of accessed information objects, i.e., they do not consider a structure like tasks or activities that orchestrates the object access.

  The TimeScape Desktop [222] provides a time-centric approach to information retrieval directly integrated in the desktop metaphor. The idea is that users actively maintain their TimeScape Desktop, i.e., they remove items they do not need for their work. Later they are able to go back in time to access earlier desktop states. The date selection can be achieved by a timeline, a calendar view and a keyword search. The active maintenance of the desktop is fundamental for the usefulness of the system.

  The CAAD system [217] supports work awareness by visualizing clusters of information objects used during the work process.

- **Required Data:** To visualize awareness information, the listed tools monitor the user system interaction to create interaction histories.

- **Addressed tensions:**
  - *Forget tasks (T1):* The review of earlier work helps the information worker to remember started tasks.
  - *Forget task status (T2):* Next to the role for prospective memory, the visualization of the work process also helps to remember the status of different work tasks.
  - *Maintain active system (T3):* Although the information objects are not in the context of tasks they belong to, a list of information objects ordered by the time they were focused is a good support for the information worker to access relevant elements.
  - *Maintain multiple active activity systems (T4):* When an information worker works on different tasks in parallel, the visualization of the work process helps in regaining an overview of the executed work.

- **Solution effects:** The described solutions do not directly address the demand to maintain an overview of activity execution. The maintenance of information about time spent with information objects given by Social Wakoopa, Rescue Time, PersonalVibe, the TimescapeDesktop, Feldspar and the Outlook Journal is not related to the activities. The individual needs to remember which activity involved which information to get the overview. CAAD provides clusters of information objects that represent activities. This provides an idea of connected objects. Still, CAAD does not give an idea of the actually performed interactions on the objects, the connections between the information objects within the activity and the respective relevance of an information objects for an activity. Therefore, all reviewed solutions have the advantage of requiring little user interaction but they complicate the encoding of the information which emerges as a new tension between information worker, task context, the interaction visualizing tool which is complex to read, the workflow which remains unclear and the object.

5.3.3 Activity Specific Information Object Access (Act-IO-Accss)

The tool has information regarding the activity a user is working on at a specific point in time. Based on this information, the tool improves the access to information objects that are described as activity related. Information seeking is an important and extensively discussed problem in information work [42, 156]. Generally, information seeking is supported by recommender systems that focus on relevance as the semantic relatedness of a search query to an information object [173]. A complementary approach focuses on the use of activity information as input for information searches to identify objects based on activity relatedness [4].

- **Solutions:** To realize activity data based recommendations, a system must have information about user tasks and information objects. Two different approaches can be distinguished. The first approach assumes, that—although user tasks are executed weakly structured—the tasks and the related information demand is known. The second approach assumes that nothing is known about the user tasks.

  Assuming that the tasks are known, the following solutions exist. The Dyonipos system [172, 216] identifies the user task and provides information about related documents, people and locations from the user's personal and the organizational information stores. The APOSDLE system analyzes user work and identifies documents related to the activities of the user based on a distinction of navigational goals, information goals and transactional goal [161]. The TaskPredictor extension allows the TaskTracer system to identify tasks automatically, without forcing the user to explicitly identify which task he is working on [258]. The task information for the Dyonipos and the APOSDLE system is generated in a machine learning process before the system is actually used. TaskTracer collects data about existing tasks and activity switches based on user input while the system is used. Middleton et al. developed the Quickstep and the Foxtrot system [186]. The system creates interaction histories for the access of research papers and uses the IBk [6] classifier to determine a paper class a research paper belongs to, which is added to an ontology.

  If there is no information about work, the detection requires methods like activity mining. The CAAD system [217] performs an activity mining on interaction histories. Activities are represented as collections of information objects. Activity identification is used to select the active activity while a user is working and to propose the related information objects. Next to suggestions to increase information object access, the collections also provide an awareness of the performed activities. Similarly, the activity based search system creates graphs of user system interactions and assumes that the task structure is inherently included in the graph [107]. Based on the activity data, the information retrieval process uses the task context to calculate relevance.

- **Required Data:** The system requires an interaction history that represents the user interaction at a certain point in time. The most recent elements of the interaction history are compared against a collection of information about activities. Such collections can be created for initially identified activities (e.g., APOSDLE and Dyonipos), training data generated during the activity execution (TaskPredictor) or based on plain interaction histories (Activity Search System and CAAD).

- **Addressed tensions:**

  – *Maintain active system (T3):* The tools proactively recommend information objects related to the task an information worker executes, which solves the tension.

- **Solution effects:** The described methods follow different approaches to maintain the information about existing activities. All training based methods require substantial effort to generate the training data. The initial training (Dyonipos and APOSDLE) in particular requires information about existing activities which is not likely to exist for the information work considered in this document.

5.3.4 Activity Specific Interruption (Act-Intrpt)

Data forwarding to the user is filtered based on performed activities.

- **Solutions:** A specific type of process support addresses the identification of interruptibility. Interruptibility is the identification of ideal breakpoints to interrupt information workers. The general idea is to block information from the user as long as he is involved in complex tasks. Therefore, the system has to decide on the user attention [226] and calculate the cost of an interruption. Extensive monitoring of the user is required within these systems. Examples are Attention-sensitive Alerting [126] and the Oasis system [134, 135].
- **Required Data:** The systems require detailed interaction histories to reason about the interruptibility of the user.
- **Addressed tensions:**
  - *Interruption (T6):* The identification of the best moment to perform an activity switch resolves the interruption tension at least with respect to computer based external interruptions.
- **Solution effects:** Some interruptions are relevant and required. None of the reviewed systems considered relevance of interruptions, therefore, the systems might filter relevant information, thus generating new intra-model tensions within those activity systems addressed by ignored interruptions.

5.3.5 Information Work Information Model (IW-InfMod)

Information models are created that simplify the retrieval of existing information and guide the access of new information.

- **Solutions:** One approach direction of structured information access is given with semantic desktops. Semantic desktops organize information objects in a linked structure that represents the concepts and relations of the information worker. Examples are the IRIS Semantic Desktop[56] created in the CALO project and the Nepomuk Social Semantic Desktop [103].

  Less structured approaches focus on information extraction without a complex data scheme. The ICARUS system extracts information included in accessed emails [165]. The Suitor system extracts information from different data sources that seems to be related to the user activity [169]. Iolite (Intelligent On-Line Inferencing for Text and Email) identifies relations between information objects based on monitoring of user activity and social network analysis for emails. The relations are offered in software clients (e.g., Microsoft Outlook add-in) based on the object selections within the client [228].
- **Required Data:** The systems create representations of users, the information they own and their information needs. These representations are based on combinations of manual effort and heuristics that reason about accessible information objects.
- **Addressed tensions:**
  - Maintain active system (T3): The semantic desktops simplify information object access, avoiding search tasks due to a large amount of information involved in a task.
- **Solution effects:** The semantic desktop tools provide a general perspective of information an information worker interacts with and unfold the included relations. Therefore, they represent powerful, relation driven, information retrieval tools. The creation of relations that belong to activities need to be maintained manually which means a substantial effort, comparable to the effort of maintaining activity planning tools. Tools like Suitor and Iolite do not model activities, therefore, the proposed information is not related to the user's work process but only considers overall interest tendencies. Therefore, generated proposals may generate irrelevant interruptions.

System	Act-Pln	Act-Awrns	Act-IO-Accss	Act-Intrpt	IW-InfMod	Soft sensor	Maintai-effort	A-priori
○ Outlook Task List [203]	x						+++	no
○ Lotus Activities [133]	x						+++	no
○ Remember The Milk [130]	x						+++	no
○ TV-ACTA [23]	x						+++	no
○ Task Assistant [209]			x				+++	no
○ TaskNavigator [124]	x		x				+++	partly
○ Kasimir [243]	x		x				+++	no
○ Sphere Juggler [190]	x					x	++	no
○ UMEA [143]	x					x	++	no
○ Periphery Shrink [225]	x						+++	no
○ 3D Window Manager [224]	x		x				+++	no
○ Rooms [117]	x						+++	no
○ Co-Activity Manager [128]	x						+++	no
○ Groupbar [264]	x		x				+++	no
○ Activity Based Computing [20]	x		x				+++	no
○ Giornata [287]	x		x				+++	no
○ Outlook Journal [30]		x					+	no
○ Rescue Time [131]		x					+	no
○ PersonalVibe [37]		x				x	+	no
○ TimeScape Desktop [222]	x	x	x				+++	no
○ Feldspar [53]			x				+	no
○ Social Wakoopa [129]		x					+	no
○ TaskTracer [77]	x					x	++	no
○ Dyonipos [172, 216]			x			x	+	yes
○ TaskPredictor [258]			x			x	++	yes
○ APOSDLE [161]			x			x	+	yes
○ CAAD [217]	(x)	(x)	x			x	+	no
○ Activity Based Search System [107]			x			x	+	no
○ PPATM [299]	x					x	+	yes
○ Attention-sensitive Alerting [126]				x		x	+	no
○ Oasis System [134]				x		x	+	no
○ Suitor [169]					x		+	no
○ IRIS [56]					x		++	no
○ Nepomuk [103]					x		++	no
○ Iolite [228]			x		x		+	no

Table 5.2.: State of the art to functionalities and information demand (Maintai-effort = Maintenance effort, A-priori = required a-priori knowledge).

5.3.6 Intermediate Results

The state of the art review has given an overview of different types of information work support tools that focus on individual work execution. An overview of the collected information is provided in Table 5.2.

The review has shown that the tools address many of the identified tensions existing in the information work heterarchy as provided in the previous section. Especially task management has presented itself as a useful approach which covers many tensions. A problem for most task management solutions is the substantial user effort required to maintain an externalization of the activities and involved elements. Process awareness is another promising approach which is limited by the considered data within the reviewed systems (information access times are displayed but not mapped to the activities they are related to). Recommendation tools for information objects and processes also require extensive manual maintenance effort to make the system useful for the work processes. Additionally, some systems are build on the crucial assumption that much information about work activities and related work processes exists a priori (during system design time).

The assessment shows that the collection of activity data without manual effort and representations that focus on human activity externalizations like tasks are relevant aspects which should be considered in support tools. Assumptions about occurring activities beforehand are critical and should be avoided.

The collected information informs the requirement specification based on ASM heterarchy transformation in the next section.

5.4 Requirements Specification II: Requirements

The second step of the requirement specification applies the process of ASM heterarchy transformation specified in the previous chapter (see section 4.5). The transformation strives to dissolve the tensions related to memory failures in information work identified in this chapter (see section 5.2 and Table 5.1). The transformation process is informed by the benefits and deficiencies of existing system designs identified in the state of the art review conducted in the previous section (see section 5.3). Based on the modifications respective requirements for software to address memory failures are specified (see section 5.4.5).

The requirements identification based on AT-SDM is a six step process (see section 4.5.2). The first steps comprise the selection of the first parent node to be modified and the selection of a modification perspective by choosing a mediator. Here, the parent system is the task system which might exist multiple times in parallel. The tool perspective is chosen, i.e., tensions need to be resolved by modifications of the tool mediator which result in requirements for tool development.

Different strategies to address the tensions have been discussed within a group of researchers to address the failures directly within the task ASMs they emerge from (e.g., modification of the used tools or the modification of context elements). However, the final approach followed focuses on the introduction of compensatory ASMs of low complexity. The tension analysis already showed that tensions are frequently addressed by compensatory ASMs which have a high complexity. For example forgotten information objects result in additional search activities (cf. section 5.2.5). The idea is to offer compensatory activities of lower complexity to address the memory failures.

In the following, the modifications of the ASMs with the tensions are reported. The modifications provide further information about the actual structure of the introduced compensatory activities. Especially it is necessary to consider that the introduction of an activity does not generate unacceptable new tensions in the system.

5.4.1 Address Tensions I: Multitasking

For the multitasking work situation two tensions have been identified. On the one hand the threat of forgotten activities due to the limited awareness (T1). A tension captured by the questions “What did I plan to do?” On the other hand the problem of maintaining an overview of the elements which belong to several active ASMs (T4). A tension addressed by the questions “Where do the things belong to?”

To address these tensions a tool is introduced which gives access to a compensatory activity. The activity offers an overview of all existing activities and shows which information objects and applications belong to which activity (see Figure 5.8).

To avoid new tensions, the activity needs to be simple, embedded in the work environment and it needs to provide the data without effort:

- **Addressed tension:**
  - Forget tasks (T1). The subject fails to remember tasks and subordinate ASMs.
  - Maintain multiple active activity systems (T4).
- **Modification:** The tool element of each task system is extended by a support tool. The tool gives access to structured externalizations of the work process to improve the subject’s awareness of the performed activities and the involved elements.

- **Emerging system:** The emerging system is a tool function which provides access to a visualization of externalized work data as an activity overview.

  Improving awareness of work based on externalization is similar to the task management systems discussed in the state of the art. The important aspect is that the creation of a maintenance activity for the work externalization is avoided. There is no system to address manual maintenance of the data. The maintenance is embedded into the access of the work externalization, i.e., when a user decides to strengthen the knowledge of the work process, the required information is automatically displayed. This can be compared to the CAAD approach which automatically provides collections of objects which stand for work activities.

- **Resulting requirements:**
  - **RQ1:** The system should help derive existing activities (Tension T1).
  - **RQ2:** The system should help derive activity related elements (Tension T4).
  - **RQ6:** The system should use data about the information worker's work process (required by RQ4), existing activities (required by RQ1, RQ5), connections between activities (required by RQ3) and the involved elements (required by RQ2).
  - **RQ7:** The data should be collected unobtrusively to assure that the data collections requires few user effort.
  - **NF-RQ1:** The use of the system should be simple, easy to learn and quick.
  - **NF-RQ2:** The system needs to be seamlessly integrated into the computer system of the user in order to be accessible during each activity.
  - **NF-RQ3:** The system needs to operate efficiently to have a good user experience because it will run permanently.

5.4.2 Address Tensions II: Underspecified Work Process

The underspecified work process triggers different activities which originate from the subject's task activities. Such an activity which originated from a task can have a high complexity which results in an increased awareness of the activity. If the awareness of the origin task is low, the generated activity can separate from the origin and become a task of its own (see Figure 5.9).

The memory of forgotten relations is refreshed by an explicit visualization of the relations between activities:

- **Addressed tension:** Separating knowledge actions (T5). A knowledge action separates from its origin and becomes a task.

- **Modification:** Each activity includes an information work support tool which gives access to an activity of accessing relations between activities. By accessing the visualizations, the subject should be able to recall the dependency between the activities. Although a knowledge action might have become separated to address a complex need, the subject recalls the required outcome to execute the original trigger activity (see Figure 5.9).

- **Emerging system:** A tool has the object of a visualization of relations between elements. The outcome is that the subject directly understands the relations. Like for the other activities, the maintenance of these relations needs be realized without additional maintenance activities to avoid new tensions in the heterarchy.

- **Resulting requirements:**
  - **RQ3:** The system should help derive connections between activities (Tension T5).
  - **RQ6:** The system should use data about the information worker's work process (required by RQ4), existing activities (required by RQ1, RQ5), connections between activities (required by RQ3) and the involved elements (required by RQ2).
  - **RQ7:** The data should be collected unobtrusively to assure that the data collections requires few user effort.
  - **NF-RQ1:** The use of the system should be simple, easy to learn and quick.
  - **NF-RQ2:** The system needs to be seamlessly integrated into the computer system of the user in order to be accessible during each activity.
  - **NF-RQ3:** The system needs to operate efficiently to have a good user experience as it will run permanently.

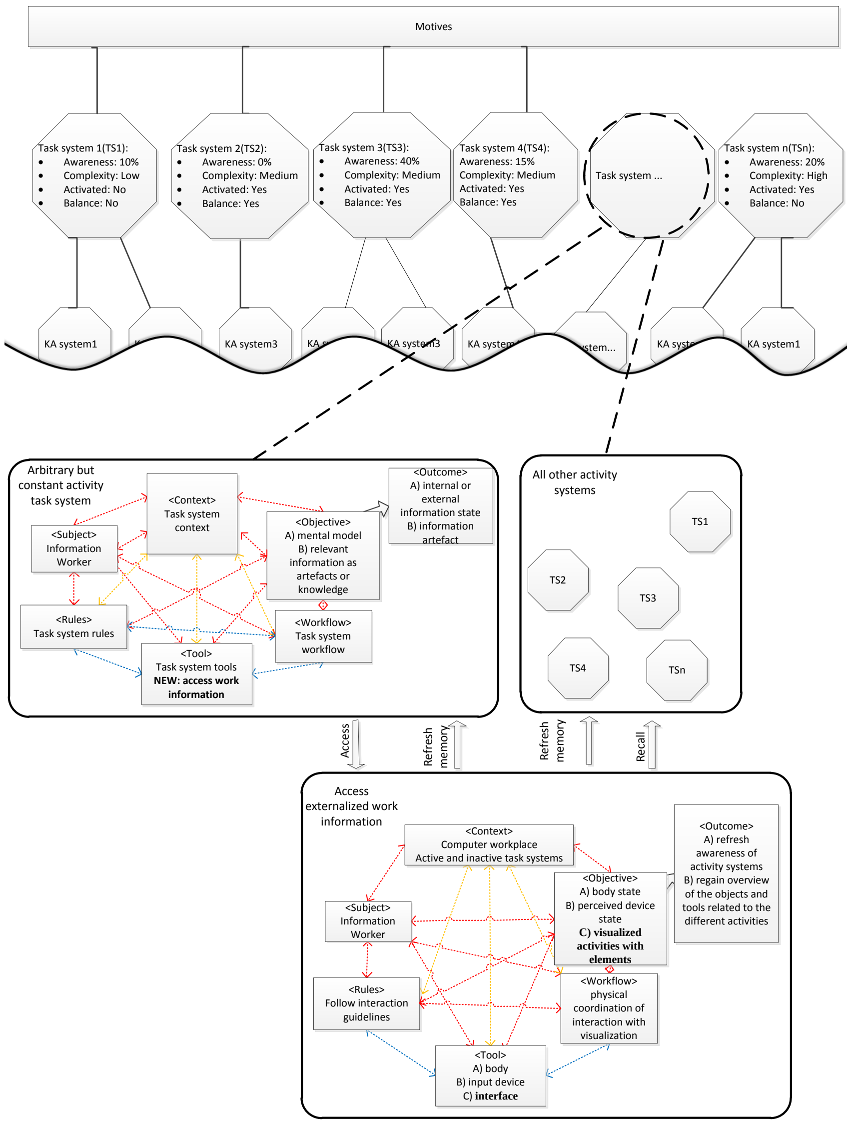

Figure 5.8.: Address tensions T1 and T4: The lack of overview and the threat of forgotten activities is addressed by a compensatory activity “Access externalized work information”. The compensatory activity is provided by a tool which is accessible from each active activity system. Therefore, the tool needs to be part of the toolset used to execute the activity. The “Access externalized work information activity” has an interface which provides access to a visualization of activities and activity related elements. By accessing the visualization the subject’s awareness of activities and involved elements is refreshed.

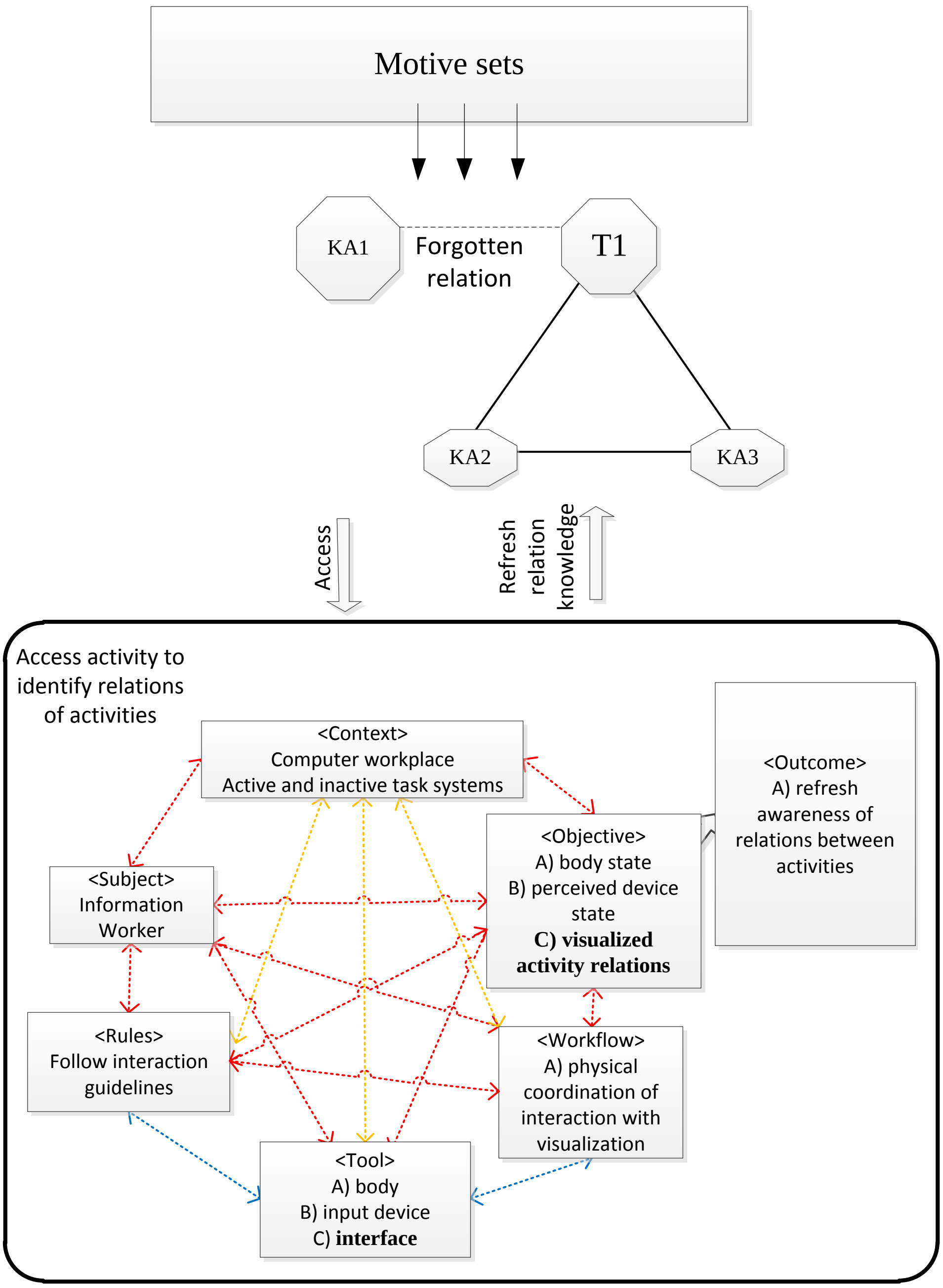

Figure 5.9.: Address Tension T5: The relation between a task and a knowledge action has been forgotten by the subject. An additional activity gives access to the relations between existing activities. The activity helps the subject to refresh the memory of relations between existing goals and respective activities.

5.4.3 Address Tensions III: Task Related Tensions

The tensions which emerge for single tasks are specializations of the tensions which emerge for multitasking (T1, T4). First, while multitasking is threatened by a loss of overview, the loss of overview can also emerge on a task level. Second, while the subject may lose the overview of the planned activities, the subject may forget the status of a single task (see Figure 5.10).

The questions that needs to be addressed are "What belongs to the activity?" and "What did I do while work on the activity and how can I continue the work?" This is again addressed by a compensatory activity which provides information focused on the single task: the work process of the task and the involved objects and tools are provided.

- **Addressed tensions:**
  - Forget task status (T2). The subject fails to remember the status of a postponed task execution
  - Maintain active system (T3). The subject fails to access the mediators and objects of the active activity
- **Modification:** The added information worker support tool gives access to an activity to access an externalization of the work process as object (see Figure 5.10). The outcome is the simplified access of required elements and their relations.
- **Emerging system:** The interaction with a visualization of the work process helps the user to understand the connection between activities and executed operations better and to regain an overview of the overall work process. Means to interact with the visualization are important to access required information.
- **Resulting requirements:**
  - **RQ1:** The system should help derive existing activities (Tension T1).
  - **RQ4:** The system should help derive executed work processes (Tension T2, T3, T4).
  - **RQ6:** The system should use data about the information worker's work process (required by RQ4), existing activities (required by RQ1, RQ5), connections between activities (required by RQ3) and the involved elements (required by RQ2).
  - **RQ7:** The data should be collected unobtrusively to assure that the data collections requires few user effort.
  - **NF-RQ1:** The use of the system should be simple, easy to learn and quick.
  - **NF-RQ2:** The system needs to be seamlessly integrated into the computer system of the user in order to be accessible during each activity.
  - **NF-RQ3:** The system needs to operate efficiently to have a good user experience as it will run permanently.

5.4.4 Address Tensions IV: Interruptions

Interruptions intrude active work processes based on a memory or an event. The subject needs to answer the question "What is this related to?" and identify an activity the memory or event relates to (see Figure 5.11).

The compensative activity which alleviates the tension helps to identify an activity based on the intrusive element which caused the interruption. Once the activity is identified the related objects and tools are accessed.

- **Addressed tension:** Interruption (T6). An interruption triggers the identification of an interruption target system and requires an activity switch.
- **Modification:** The information work support tool added to the tool list of each task mediates activity switches based on two systems (see Figure 5.11).
- **Emerging system:** The two introduced systems are described in the following. The first system has the goal of identifying an activity system that is related to the interruption. The context is given with the existing computer workspace with open applications and accessed information objects (this is already addressed by system 1).

  The second system has the goal of activating the identified activity system. The activation is a combination of cognitive and physical processes. Cognitive, as the individual needs to remember all things involved in the system and the respective elements need to be accessed. Physical, as the subject needs to interact with the computer to access the recalled elements.

  Remembering and accessing are closely connected, as an access changes the perception and helps to remember other related things. The activation process is supported by a visualization of the interruption related activity system (addressed by system 2).

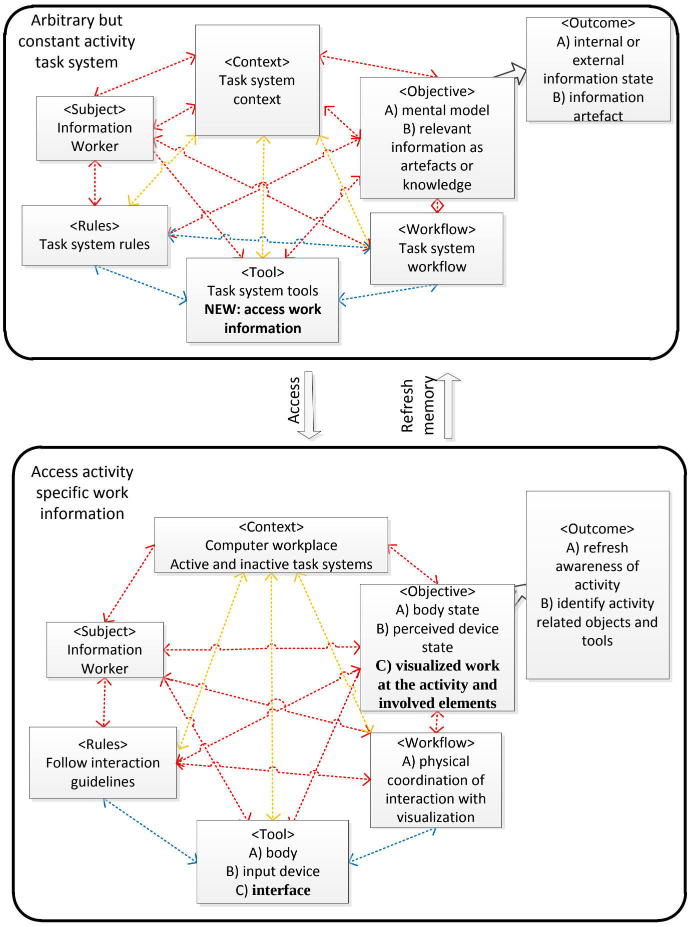

Figure 5.10.: Address tensions T2 and T3: A lack of knowledge about the status of a task and the involved information objects and tools is addressed by a compensatory activity "Access activity specific work information". The compensatory activity is provided by a tool which is accessible from each active activity system. Therefore, the tool needs to be part of the toolset used to execute the activity. The "Access activity specific work information" has an interface which provides access to a visualization of activities and elements. By accessing the visualization the subject's awareness of specific activity and involved elements is refreshed.

- **Resulting requirements:**
  - **RQ5:** The system should support activity switch as identifying and activating an activity system (Tension T6).
  - **RQ7:** The data should be collected unobtrusively to assure that the data collections requires few user effort.
  - **NF-RQ1:** The use of the system should be simple, easy to learn and quick.
  - **NF-RQ2:** The system needs to be seamlessly integrated into the computer system of the user in order to be accessible during each activity.
  - **NF-RQ3:** The system needs to operate efficiently to have a good user experience as it will run permanently.

5.4.5 Modification Based Requirement Elicitation

The previous section has applied different modifications of the ASM heterarchy of the information worker to address the identified tensions. Each modification has resulted in the introduction of one or more activities the information worker can execute. Based on the modifications and the introduced activities requirements for an information work support tool can be derived (see last step of the described process in section 4.5.2).

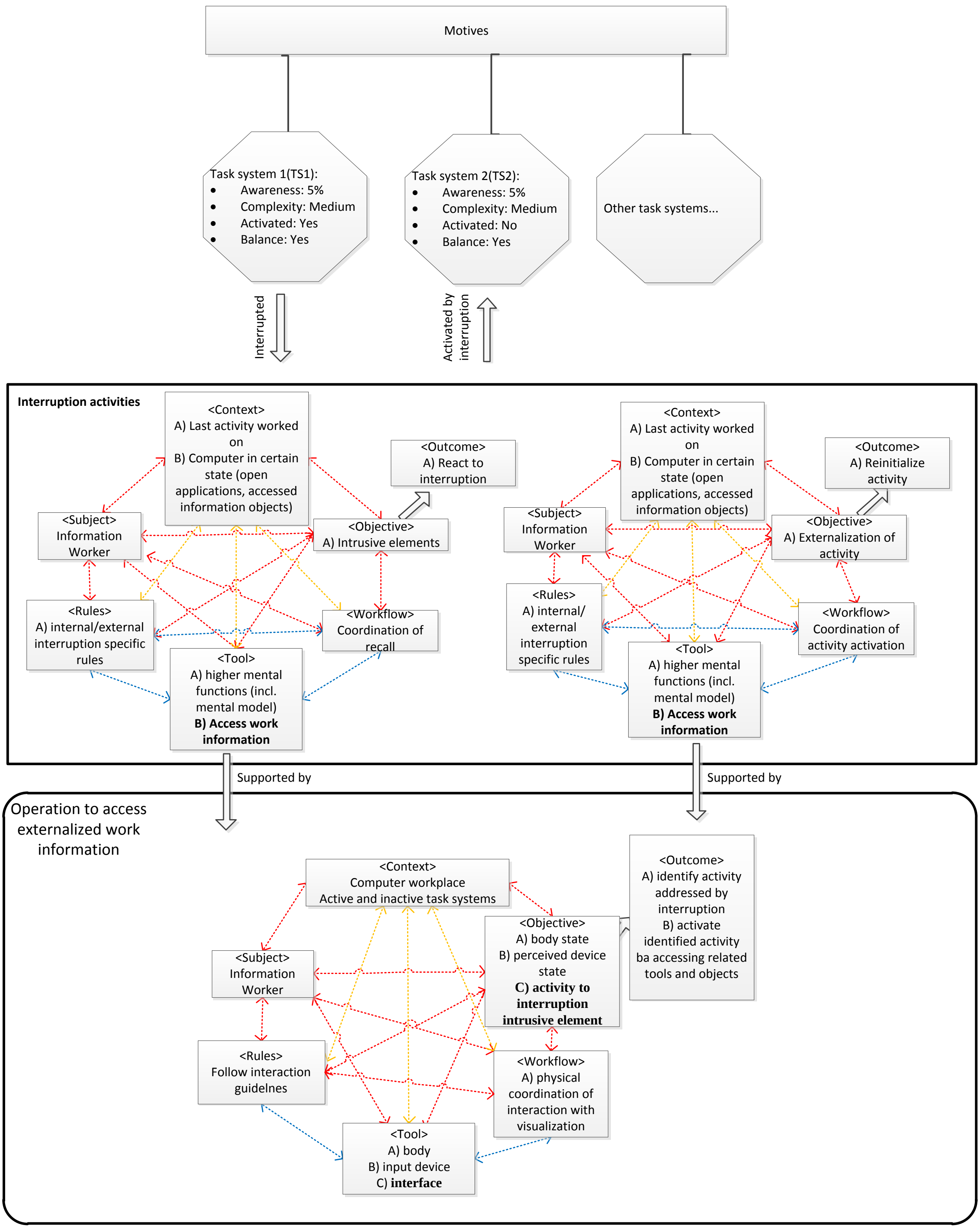

Figure 5.11.: The interruption triggers the identification of a respective activity and the activation of that activity. This is supported by an additional activity which simplifies the identification and the activation of the activity.

Each modification with attached activity stands for a functional requirement:

- **RQ1:** The system should help derive existing activities (Tension T1).
- **RQ2:** The system should help derive activity related elements (Tension T2, T3, T4).
- **RQ3:** The system should help derive connections between activities (Tension T5).
- **RQ4:** The system should help derive executed work processes (Tension T2, T3, T4).
- **RQ5:** The system should support activity switch as identifying and activating an activity system (Tension T6).

All functional requirements are closely related to the externalization of information about the work process on different levels of granularity and classified differently. Therefore, the collection of data is of specific relevance for the system. The scope of this data, its creation and its maintenance need to be addressed by additional functional requirements:

- **RQ6:** The system should use data about the information worker's work process (required by RQ4), existing activities (required by RQ1, RQ5), connections between activities (required by RQ3) and the involved elements (required by RQ2).
- **RQ7:** The data should be collected unobtrusively and require little maintenance effort by the user.

The activities provided by the information work support system (RQ1-RQ5) based on data which follows the identified requirements RQ6-RQ7 should be accessible from every activity. Still, the activities need to be executed quickly and without much cognitive effort to avoid interruptions based on the tool usage. Therefore, different non functional requirements need to be considered:

- **NF-RQ1:** The use of the system should be simple, easy to learn and quick.
- **NF-RQ2:** The system needs to be seamlessly integrated into the computer system of the user in order to be accessible during each activity.
- **NF-RQ3:** The system needs to operate efficiently to have a good user experience as it will run permanently.

As the activity data collected by the system contains important personal information, the system additionally needs to protect the privacy of the user.

- **NF-RQ4:** The system should protect the privacy of its users by preventing the misuse of the activity data.

The given requirements will be the foundation for the design of a support system for the information worker. The remainder of this thesis follows the UCD cycle and discusses realization methods.

5.4.6 Intermediate Results

The reported transformation process has shown that it is important to anticipate the effects of a newly introduced system on the given activities as far as possible.

This section has identified requirements based on the modifications of the information work heterarchy, addressing the identified tensions on the one hand and considering the state of the art in information work with a focus on the support of mnemonic processes on the other hand. Four compensating activities were introduced which refresh the subject's memory and facilitate object access to address the six identified tensions.

To realize the activities, five functional requirements to realize the activities, two functional requirements regarding the data to enable the intended support and four non-functional requirements regarding the user experience and the privacy protection have been identified. The identified requirements show a benefit of requirement specification based on the AT-SDM: the requirements support activities which are embedded in the work situation they address. Thus, constraints of the work situation like the quick access of information or the need to limit the user effort derive directly from the analysis of the ASM heterarchy.

5.5 Summary

The first two steps of the UCD-cycle using the AT-SDM method to analyze the context of use and elicit requirements have been conducted in this chapter. Based on the information work ideal type (see section 3), the context of use was created as heterarchy of ASMs. The identified heterarchy describes different information work related motive sets and proposes a structure of task, knowledge action and desktop operation activities subordinate to the motives of an information worker.

Based on the heterarchy four important work situations have been modeled. The situations have been input for a tension analysis. The tension analysis identified several tensions related to memory failures. All tensions consider the recall of activities and activity related information as well as the access to activity related tools and information objects. Tensions exist with respect to the recall of different activities and their details (forget tasks, T1 and forget task status, T2), the maintenance of one or more active activities (T3, T4) the access of required information objects, the maintenance of relations between activities (separatist tendency, T5) and interruption handling (T6).

A state of the art analysis of information work support tools with a focus on supporting mnemonic processes has shown that all identified tensions are addressed by existing solutions. Yet, by addressing all tensions they are not solved. The described methods have disadvantages as they require extensive manual effort to provide information work support, resulting in additional interruptions of the work. Additionally, some tools make the doubtful assumption that all activities are known during the design time of the support tool, resulting in very limited support capabilities from the perspective of the information work ideal type.

Based on a structured process of tension relaxation compensating activities have been specified. The taken approach of compensating activities focuses on one basic idea which is the guiding principle of the remainder of this dissertation: externalized activity data is collected and offered to the subject to facilitate the recall of information and to simplify the access of relevant tools and objects.

The remainder of this thesis provides methods to address the identified requirements. The first challenge is to identify what the term activity data actually refers to, how it is structured and how activity data is collected. This is addressed in the chapters 6 and 7. Once the challenge of activity data collection is addressed, a design space for information work support methods based on activity data is specified and respective methods are created and evaluated (see chapters 8 and 9).

Part III.
Information Work Support Tool

6 Modelling and Collecting Work Execution Data

This chapter introduces the ContAct monitor [238] as an approach for the unobtrusive externalization of interaction histories and the extraction of information relevant to understand the work process of a user. The collection of this data is a fundamental requirement to address memory threats in information work (addressed requirements: RQ6-7).

The chapter is structured as follows. First, a background on monitoring is given by describing interaction data management (see section 6.1). The remainder of the chapter iterates through a three step process of interaction data management. The second section describes the interaction data collection (see section 6.2). The third section provides an overview of the interaction data processing (see section 6.3). The fourth section describes the formalization of interaction data (see section 6.4). Finally, related work is considered with respect to interaction monitoring for user support and the formalization of information work.

The contributions for this thesis are the distinctive features of ContAct. First, ContAct captures very different types of system interaction events. The captured data includes data collected from a broad range of applications and includes the "Cont(ent)" accessed when the user performs "Act(ions)". Connecting content information with interactions increases the value of the interaction history, as the data enables a topic based analysis of the work. Second, a complex event processing and heuristic based approach to identify desktop operations and knowledge actions is presented. Third, ContAct provides a formalization of the resulting data in the so called computer work ontology (CWO) ontology. The CWO [242] is an extension of the DOLCE upper ontology with a focus on the computer workplace and information work execution.

6.1 Interaction Data Management

The term interaction data management describes a three step process involved in the work with interaction data: interaction data collection (see section 6.1.1), interaction data processing (see section 6.1.2) and interaction data organization (see section 6.1.3). The process is visualized in Figure 6.1. This process is implemented in most monitoring systems which process interaction histories, although it generally is not highlighted explicitly. Examples are the Swish system [204] with a desktop title sensor, a title clustering and a storage, the CAAD system [217] with an application and file sensors, processed by an algorithm comparable to GaP [44], or the systems based on the UICO module [214] which monitors a broad range of interaction data, derives higher level information comparable to desktop operations and stores them in an ontology [215].

6.1.1 Interaction Data Collection

Talking about interaction data implicitly means talking about a set of events following an event representation notation. Generally, an event is a significant change in the state of the universe [52]. Following this general definition by Chandy, a specification for interaction events shall be given. An interaction event is a state change that stands for interactions of an actor with a specified object in an environment. The following descriptions hold for the definition:

- **Actor:** The actor triggers an interaction. An actor can be a system, a natural person or a group of natural persons.
- **Object:** The object is the thing an actor interacts with. The object can be any physical entity.
- **Interaction:** An interaction is any kind of act[1] that is executed by an actor towards an object. The act does not necessarily need to be reasonable but may be accidental or unplanned.
- **Environment:** The environment is a spatial or logical limitation for the interactions that are considered as relevant. The environment can be a building or a country, as well as a computer system or even an application.

To classify interaction data management of a system, it is necessary to describe the object, actor, interaction and environment for the interaction events managed by the system.

Data providers deliver information related to the favored interaction data. The providers observe the environment interactions take place in and generate basic events as discrete instances of a (possibly continuous) signal [120]. Observation of the environment may be realized based on physical sensors (e.g., temperature), virtual sensors (e.g., software hubs to monitor keyboard interaction) or information streams (e.g., RSS feeds created by a group of actors). All events need to follow a similar standard that should comprise a timestamp and descriptive parameters. General rules that apply are [120]:

- A timestamp may be just one point (point semantics of time) or an interval (interval semantics of time).
- Parameters may be absolute or deltas relative to older reference values.
- Each basic event may contain additional attributes.

The resulting events are input for the interaction data processing.

[1] An act could be an activity, an action or an operation.

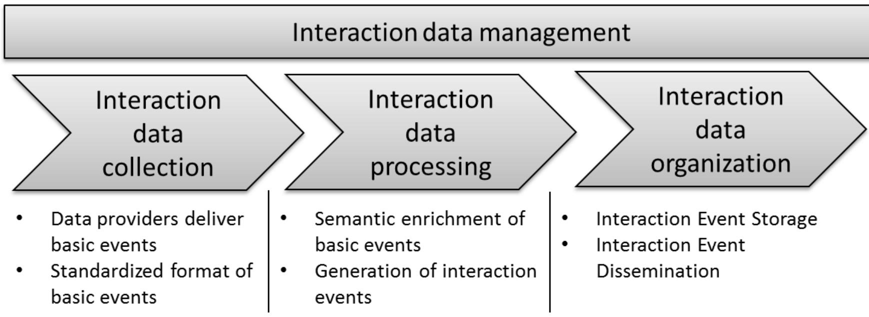

Figure 6.1.: Interaction data management processes.

6.1.2 Interaction Data Processing

Processing of interaction data is required if a gap between the data collected by sensors and the desired information exists. Interaction data processing aggregates and semantically enriches the basic events delivered by the data providers. Aggregation refers to the combination of the data collected from different sensors or from sensors over time to derive additional information. Semantic enrichment refers to the combination of the event with additional data from other data sources. Often aggregation and semantic enrichment are closely combined during data processing.

A simple example is a speed measurement which is used to check whether one drives too fast. The question can only be answered by combining information about the actual speed and the allowed speed.

Generally, the technique that enriches the events with additional semantics needs to be identified case by case. Complex event processing [166] is one suitable method to process the event streams, especially if rules can be generated to build interaction events based on the basic events delivered by the data providers. Another example is classification based on models trained by machine learning.

6.1.3 Interaction Data Organization

The organization of interaction data as the final step comprises the data storage and data dissemination. The storage of interaction data provides access to interaction histories and enables the ex post analysis of the data. Databases or structures like XML and ontologies are frequently used (e.g., APOSDLE uses database and XML [161], UICO is an ontology used to capture the data [215]). Dissemination addresses the access to stored interaction data as well as the instant forwarding of data within the system or to subscribers.

6.2 ContAct: Interaction Data Collection

Following the scheme and process description given in the upper section, ContAct is an interaction data management system that identifies interactions at the computer workplace. The system logs information of a user who is considered to be an information worker (*actor*). The user interacts with a computer ContAct is running on (*environment*). An interaction is triggered by a foreground window change, a mouse click and keyboard input and addresses applications and information objects (*object*). The ContAct system is an extension of the APOSDLE monitor application [161]. The system has been implemented as a service that can be subscribed by other applications to use the extracted interaction histories.

The computer workplace is mediated by the operating system that organizes system input and output. System monitoring benefits from the mediating role of the operating system. Sources for monitoring functionalities may be frameworks to organize data exchange between applications (e.g., the interoperability libraries for windows) and accessibility features (e.g., the UI Automation Framework for windows). This considers the recommendations for monitoring applications given by Fenstermacher et al. [87] and extends them with respect to accessibility frameworks as newly accessible source of information. The ContAct monitor combines information from the following sources:

- **Input and Output Devices**
  - *Mouse:* Mouse movement, mouse wheel operations and mouse clicks are captured by a mouse hub.
  - *Keyboard:* The Keyboard input stream is captured by a keyboard hub.
  - *Printer:* Printed documents are captured.

- **System Management**
  - *Process:* Information about running processes can be accessed, partly standing for applications. For each process detailed information can also be accessed, like windows belonging to a process and files that are locked by a process. The most valuable information is the focus window, identifying the process a user is actively working with.
  - *Filesystem:* The files accessed during the work of a user, including information like the creation and modification date. ContAct explicitly checks for the files locked by the running programs, parses their content and enriches the process information with this file data.
  - *Clipboard:* The clipboard is used to copy and paste different types of data within or between applications with respective information objects.

- **Accessibility Features**
  - Accessibility features provide access to data structures that represent all running instances and potentially visible elements that exist at a specific point in time on the running system instance. An example is the UI automation framework which provides a tree representation of the user desktop. Each visualized element is part of the tree and may support a set of patterns which can be used to interact with the visualized element (e.g., query displayed text).

- **Special Applications**
  - Application APIs enable the subscription of the events generated by a running instance of a program.
  - Application APIs may provide access to the information objects that are displayed by the application. An example for such information objects is the content of a text file displayed by a word processor or the content of a website displayed by a web browser.


```
<event eventName="FOREGROUND_WINDOW_CHANGED" atTime="01.02.2011 16:19:26.484" eventCategory="Process"><eventattributes>
        <eventattribute name="processname" type="String" value="POWERPNT" />
        <eventattribute name="windowtitle" type="String" value="Microsoft PowerPoint - [Planning-Roundup_v3.pptx]" />
        <eventattribute name="associatedFile" type="String" value="\BaseNamedObjects\MSCTF.Shared.SFM.AFDB" />
        <eventattribute name="associatedFile" type="String" value="C:\Documents and Settings\evaluationuser3\Desktop\Work\Planning-
Roundup_v3.pptx" /></eventattributes></event>
<event eventName="FILESYSTEM_OBJECT_DELETED" atTime="01.02.2011 16:19:26.828" eventCategory="Filesystem"><eventattributes>
        <eventattribute name="name" type="String" value="C:\Documents and Settings\evaluationuser3\Desktop\Work\Planning-
Roundup_v3.pptx" />
        <eventattribute name="notifyfilter" type="String" value="FileName" /></eventattributes></event>
<event eventName="FILESYSTEM_OBJECT_CREATED" atTime="01.02.2011 16:19:26.828" eventCategory="Filesystem"><eventattributes>
        <eventattribute name="name" type="String" value="C:\Documents and Settings\evaluationuser3\Desktop\Work\Planning-
Roundup_v3.pptx" />
        <eventattribute name="notifyfilter" type="String" value="FileName" /></eventattributes></event>
<event eventName="MSPOWERPOINT_EVENT_THROWN" atTime="01.02.2011 16:19:26.968" eventCategory="Application"><eventattributes>
        <eventattribute name="title" type="String" value="Planning-Roundup_v3.pptx" />
        <eventattribute name="content" type="String" value=" Planning Demand and Master Planning Roundup Introduction Author: Andreas
Goeb&#xD;Date: 2011/02/01 Demand Planning Purpose&#xD;improve decisions affecting demand accuracy &#xD;calculation of buffer or
safety stocks &#xD;Results: &#xD;Demand Plan&#xD;Benefit:&#xD;Increased performance of each supply chain entity Master Planning
Purpose&#xD;synchronize the flow of materials along the supply chain&#xD;Range: Mid-term  At least one seasonal
cycle&#xD;Contents:&#xD;Production&#xD;Transport&#xD;supply capacities&#xD;Seasonal stock &#xD;balancing of supply and
```


Figure 6.2.: Raw sensor events monitored by the ContAct monitor application.

For interaction data collection a combination of the described sensors is used. Process information is used together with mouse and keyboard hubs to trigger data requests. Once a process changes, an enter hit or a mouse click is identified, dedicated application monitors are used together with the UI automation framework to extract information about the interaction. The extracted events include the following information:

- **Event Type ID:** Each type of event has a unique identifier.
- **Event Category ID:** Events have been organized by the categories process, filesystem, application, mouse, clipboard, keyboard, printer which have unique identifiers.
- **Timestamp:** Event occurrence time.
- **Process:** The system process, the event belongs to.
- **Attribute List:** Dependent on the event type, different attributes can be identified. This can contain information objects and content. For other information object other event specific information is included (e.g., the sender for emails).

An example excerpt from the monitored events is visible in Figure 6.2. The example shows that each monitored event contains a list of attributes, has a timestamp and belongs to an event category. The categories were introduced to organize the different generated events. The initial logging operation is performed, using an extension of the XML scheme used for the APOSDLE process. Extensions mainly address event types which were not integrated in the APOSLDE monitor: events about user interface elements the user interacted with (e.g., labels of clicked buttons).

Some types of data require complex implementations and are computationally intensive. An example is the content monitor. To cover the different types of content, users interact with a variety of computationally intensive methods needs to be implemented, including parsing the accessed file (e.g., PDF files), reading the automation elements with the automation API (e.g., accessing the display document object model in browsers), web crawling for websites or requesting the data directly from the subscribed application API (e.g., Microsoft Office API).

The set of monitored events can be configured to find a balance between the data required and the resource consumption as requested by [77].

6.2.1 Intermediate Results

The sensor based collection of interaction data has been described in detail, providing an overall understanding of the sensor elements used in the ContAct monitor. The description completes the data collection step of the ContAct monitor.

6.3 ContAct: Interaction Data Processing

In the following, the identification of desktop operations and knowledge actions from the software sensor data is described. For desktop operations rules are applied using complex event processing (CEP) [166], while knowledge actions are identified based on heuristics. Due to the partonomic relation between events, desktop operations and knowledge actions, a hierarchical extraction process results. First, complex event processing is used to detect desktop operations (see section 6.3.1). Second, heuristics are applied to the desktop operations to identify knowledge actions (see Figure 6.3 and 6.4, see section 6.3.2). The process has been implemented in the ContAct monitor.

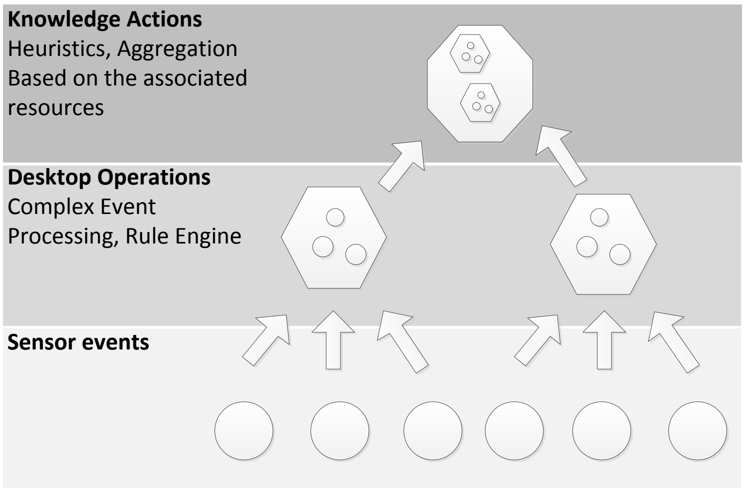

Figure 6.3.: Hierarchical and partonomic relation between knowledge actions, desktop operations and sensor events.

Rules for CEP and heuristics have been derived from the manual analysis of logged interaction histories. Sets of elements which stand for desktop operations and knowledge actions were selected and compared to identify characteristics useful for the creation of rules and knowledge actions (for this task data set 1 one was used which already included manual annotations for each task execution, see section C.1).

6.3.1 Identifying Desktop Operations

As described in section 3.3.3.1, desktop operations are simple work techniques which describe basic user system interactions. The vocabulary is closely aligned with the desktop metaphor and the WIMP paradigm (see section 3.3 and [18]).

Each desktop operation is executed by a set of interactions which resemble a workflow. Thus, desktop operation identification results from a mapping of events in the monitored event stream to the desktop operation taxonomy.

Due to the different workflows there is no unique set of events for each desktop operation. An example is saving a file which can be triggered by combining the key CTRL + S, by clicking an icon in a menu bar and by opening a toolbar element and selecting the save entry.

Desktop operations are not application specific but may be implemented in different applications. Due to the standardization of interactions, among most applications the same interaction workflows generate the same desktop operation. Example: The file save by CTRL + S or by the icon and the menu interaction workflows work in most modern applications.

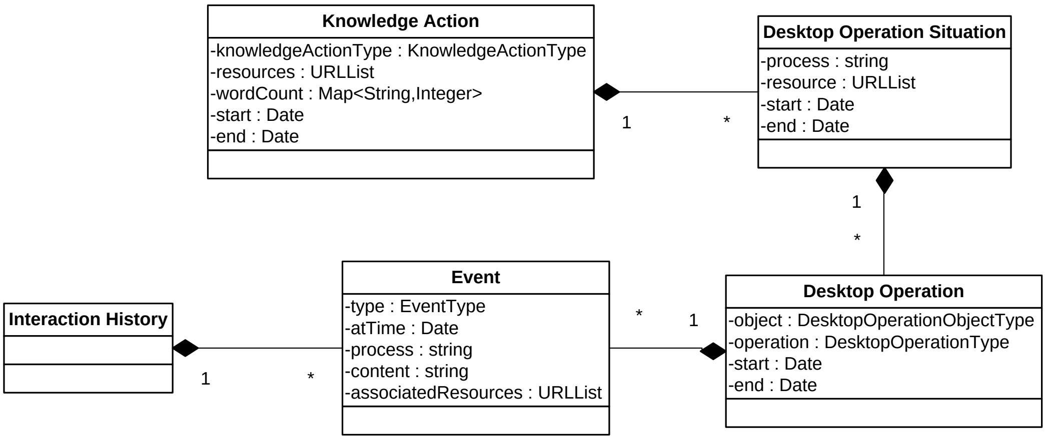

Figure 6.4.: Elements and attributes involved in the process of knowledge action and desktop operation extraction.

To cover the different approaches which lead to desktop operations, a rule based approach was chosen to identify desktop operations. The approach benefits from the similarity of workflows among different applications, decreasing the overall amount of rules. Nevertheless, the number of rules is still high. Overall, 98 rules were modeled to realize the identified set of 25 different desktop operations (see table 6.1).

The implementation of the rules in the ContAct monitor uses Drools fusion[2], a complex event processing engine that supports temporal reasoning. Temporal reasoning is important, as the temporal order of events is relevant to identify the workflows. The performed complex event processing generates desktop operation objects based on interaction history events. Desktop operation objects formalize the concept of desktop operations and, as instances, realize a perspective on a work process as a sequence of desktop operations (cf. Figure 6.4).

Each desktop operation has a begin and an end attribute. Desktop operations have been modeled as being composed of inseparable workflows, i.e., when a user executes a desktop operation he is not interrupted by another desktop operation. Considering desktop operations as inseparable refers to their operational nature: the subject performs a workflow as a trained inseparable operation. Only adaptation might be necessary to react to given conditions which have been reflected in the rule modeling (e.g., within timeframes events may occur within a desktop operation which do not belong to the modeled ideal workflow).

The resulting structure of desktop operations is the following:

- **Operation:** The object, the desktop operation is executed on.
- **Object:** The type of desktop operation performed on an object.
- **List of events:** The events which occurred to realize the desktop operation.
- **Begin:** Timestamp for desktop operation start.
- **End:** Timestamp for desktop operation completion.
- **Duration:** The time the desktop operation took.
- **Application:** Application the desktop operation was performed on.
- **Information object:** The information object which was focused in the application the desktop operation was performed on, if available.

Two desktop operations are similar if the same operation on the same object has the same begin, the same end and is performed in the context of the same application on the same information objects.

6.3.2 Identifying Knowledge Actions

Knowledge actions are work techniques a user applies in a work process to transform an object to a goal (see section 3.3). Generally, a mixture of different knowledge actions is combined to actually achieve a goal (e.g., authoring is frequently combined with consumption activities). In terms of interaction histories, a knowledge action is a disjoint unit of continuous work as the related events may be spread over the complete history. Knowledge actions are realized by desktop operations that are scattered among the interaction history. Therefore, knowledge actions do not have a start and an end time but contain a set of time periods, the knowledge action was active in. This is an important difference between knowledge actions and desktop operations with their uninterrupted interaction workflows (see Figure 6.5).

[2] http://www.jboss.org/drools/drools-fusion.html

Opr \ Obj	App	File	Folder	Information Object	Window
Open	x	x	x		
Close	x	x	x		
Save		x			
Rename		x	x		
Delete		x	x	x	
Cut		x	x	x	
Paste		x	x	x	
Send		x		x	
Create		x	x	x	
Execute	x				
Focus			x		x
Move				x	x

Table 6.1.: Desktop operations: possible pairs of operation (OPR) and object (OBJ).

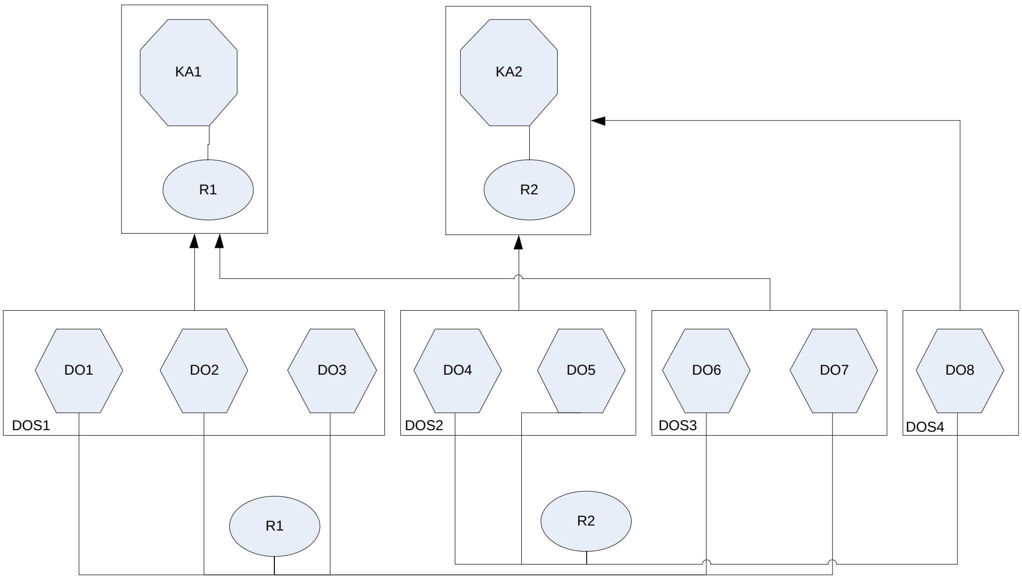

Figure 6.5.: Relation between Knowledge Actions (KA1 and KA2), resources (R1 and R2) and Desktop Operations (DO1-DO8).

Knowledge actions are generated in a two-step process. First, knowledge actions are identified. Second, knowledge actions are classified based on heuristics. Two knowledge actions are similar if they are of the same type and if they are executed on the same application and the same information objects. A knowledge action object contains the following attributes:

- **Knowledge Action Type:** The ContAct monitor considers Browsing, Authoring, Communicating, Consuming and Organizing.
- **Desktop operation:** The desktop operations the knowledge action is composed of.
- **Content:** The text accessed in the context of the knowledge action. This includes the content of the accessed information objects but also keyboard input, labels of selected buttons and other types of text the user interacted with.
- **Start:** The first activation of the knowledge action.
- **End:** The last activation of the knowledge action.
- **Activation:** A list of tuples denoting the start and end times of the knowledge action's active times.
- **Activate Duration:** The duration the knowledge action has been active.
- **Switched to knowledge actions:** The knowledge actions which were performed before the respective knowledge action.
- **Switched from knowledge actions:** The knowledge actions which were performed after the respective knowledge action.

To identify knowledge actions, those desktop operations are collected which are performed on the same application and on the same information objects (see algorithm 1). This assumes that only one knowledge action is performed on the same application and the same information object. This perspective is applied within this thesis. Still, the assumption is a simplification. The authoring activity of the same object may exist in the context of different activities. The actual relevance of the use of similar knowledge actions in different knowledge actions needs to be investigated further in future research.

```
Data: List<DesktopOperationSituation>
Result: Set<KnowledgeAction>
HashMap<KnowledgeAction> collectedKnowledgeActions;
for DesktopOperationSituation s in List<DesktopOperationSituation> do
    if ∃ k with (s.Application ⊂ k.Application ∧ s.InformationObjects ⊂ k.InformationObjects) then
        k.Add(s);
    else
        KnowledgeAction k=new KnowledgeAction(s);
        collectedKnowledgeActions.put(k);
    end
end
```

Algorithm 1: Creation of knowledge actions.

The resulting knowledge action elements are classified based on heuristics. The applied heuristics generate a value which indicates the probability that a knowledge action belongs to a knowledge action type. The type with the highest score is used for the knowledge action.

- **Authoring type:** Authoring refers to the creation of textual or other media content. Creation is a desktop operation which can be performed on files, folders and information objects. The creation of relevance for authoring is the creation of information objects: text fragments, lines in a graphic, etc. The indication value is calculated as the number of authoring related desktop operations (those with operation create and object information object) divided by the number of all non authoring related desktop operations: $aut_{ind} = c_{autcom} * \frac{|A|}{|E \backslash A|}$ with $A = DeskOp(Opr = Create \wedge Obj = InformationObject), E = DeskOp_1, ..., DeskOp_n, 0 < c_{autcom} < 1$

- **Communication type:** The considered knowledge actions focus on the individual work activities. Communication is the only knowledge action which captures the dissemination and the exchange of information from other people. Communication covers very different communication types: synchronous and asynchronous as well as voice based or text based communication.

  Text based communication at first glance can be considered as authoring. Still, the initial studies as well as the manual labeling of interaction histories showed that both are perceived differently. If a text is authored with a tool that focuses on communication, e.g., a messenger or an email tool and the authored text is sent directly to a receiver, the communication intent has more relevance than the authoring. Therefore, the occurrence of a sent desktop operation increases the communication indicator to be always greater than the authoring indicator.

  $comm_{ind} = c_{autcom} * \frac{|A|}{|E \backslash A|} + \frac{|S|}{c}$ with $A = DeskOp(Opr = Create \wedge Obj = InformationObject), E = DeskOp_1, ..., DeskOp_n$, $S = DeskOp(Opr = Send), c > 0$ and $0 < c_{autcom} < 1$

- **Organization type:** Organization as the application of organization schemes to files and information objects mainly manifests in the application of copy, cut, paste, delete and rename operations. These are basic file based organization schemes. For the considered data sets, organization schemes which were not file based were not used. Despite the focus on file based classification, generally a deeper investigation into elaborate classification and categorization schemes like tagging or ontology integration is of interest.

  The indication value is calculated as the fraction of organization related desktop operations (rename, delete, cut, copy, paste) to the other desktop operations.

  $org_{ind} = c_{org} * \frac{|O|}{|E \backslash O|}$ with $O = DeskOp(Opr = Rename, Delete, Cut, Paste, Copy), E = DeskOp_1, ..., DeskOp_n$ and $0 < c_{org} < 1$

- **Consuming type:** Consuming has been described as processing the visual representation of information. Software sensors are not very good at identifying consumption. Sensors like a camera or electroencephalography would be two approaches which provide valuable data for consumption. Considering the need to limit the intrusiveness of the data collection, an implicit approach based on software sensors was chosen. For software sensors, interaction schemes limited to operations which support the data consumption by scrolling, panning and zooming (collected by the move desktop operations) and a lack of other desktop operations can be considered as indicators for consumption. Therefore, consuming is assumed if only desktop operations like mouse movements occur.

  $cons_{ind} = c_{cons} * \frac{|C|}{|E \backslash C|}$ with $C = DeskOp(Opr = Move), E = DeskOp_1, ..., DeskOp_n$ and $0 < c_{cons} < 1$

- **Browsing type:** Browsing refers to the search for relevant information sources with respect to an information need. Browsing presents itself as a sequence of opening and focusing operations on different information objects. The process is often coordinated by one or more information sources which provide access to the browsed objects. An example is a search website which lists many search results with respective hyperlinks. The user clicks on hyperlinks, reviews the content with respect to its relevance for an information need, goes back and tries a different hyperlink.

Browsing is identified if more than two open desktop operations on files and information objects occur in short succession. The set of knowledge actions is iterated with respect to successive (i.e., uninterrupted, consisting of only one desktop operation), short consumption knowledge actions. Short consumption knowledge actions are merged to browsing knowledge actions.

The browsing knowledge action is identified substantially different than the other knowledge actions and involves merging different knowledge actions. As browsing builds on a specific type of consumption, it is identified last. For the remaining knowledge actions, the indicators are calculated and the highest indicator is used to assign a knowledge action type. The constants $c_{cons}, c_{org}, c_{autcom}$ are needed to align the results of the different calculations. The constant c_{autcom} is similar for communication and authoring to assure that a sent desktop operation results in a preferred choice of communication over authoring type.

6.3.3 Intermediate Results

Methods to identify knowledge actions and desktop operations based on interaction data logged by software sensors have been described. The methods allow the use of those concepts identified earlier (knowledge actions, desktop operations) in an applications. The description completes the interaction data processing step of the ContAct monitor.

6.4 ContAct: Interaction Data Organization

The data collected and processed by the ContAct monitor is organized using the CWO ontology. The CWO ontology provides a structured vocabulary to capture work processes at a computer. The process understanding follows the information work understanding established in chapter 3: The computer workplace is considered as a structured work execution environment which is used in ad-hoc work processes by the application of work techniques. Therefore, the described ontology tackles three aspects.

First, individual work execution is modeled in terms of knowledge actions and desktop operations (see section 6.4.1). Second, the computer workplace and the information access and modification is modeled. Applications, application scenarios and application functionalities describe the tools used. Information is structured in information objects and can be input for applications (see section 6.4.2). Third, the work processes and the work environment are connected which allows the description of individual work processes in terms of applied work techniques, goals and their manifestation by actual use of applications to create, access and/or modify information (see section 6.4.3).

6.4.1 Background for the Computer Work Ontology

To address the mentioned modeling goals ontologies have been chosen. An ontology is “an explicit specification of a conceptualization” [104], with a conceptualization as “an abstract, simplified view of the world” [104]. Therefore, the gained understanding of information work can be captured formalized in an ontology. Thereby, human readability is maintained [301]. Other beneficial aspects are modularization, reasoning and a non-hierarchical net structure which avoids favoring specific information [2]. The use of ontologies to model information work is common, considering approaches using RDF [2], RDF-S [233] or OWL [56] ontologies.

The specific focus of the computer work ontology is the integration of categories of work execution and the computer workplace. An integration to capture monitored data and derived information in the ontology, thus combining the output of the collection and the processing phase of the ContAct monitor.

CWO extends the DOLCE upper ontology. Whereas domain ontologies focus on a minimal terminological structure, upper ontologies describe general concepts which are valid across all knowledge domains. The DOLCE upper ontology was chosen as a well-known upper ontology with a “cognitive bias [of] capturing categories underlying [...] human commonsense” [96]. DOLCE has been designed as a first module of a Foundational Ontologies Library [177, 96].

As upper ontology DOLCE describes the relationship among endurant and perdurant particulars, thereby aiming at capturing the ontological categories underlying natural language and human common sense. The library contains different systematically related modules and defines different design patterns to reuse the content for more specific domains [95]. The following ontologies are reused in this section (cf. Figure 6.6):

- **Descriptions and Situations (DnS):** An ontological theory of contexts. DnS can be considered an ontology design pattern for structuring core and domain ontologies that require contextualization.
- **Ontology of Plans (OoP):** Formalization of a generic theory of plans.
- **Ontology of Information Objects (IO):** A semiotic ontology design pattern that assumes a content transferred in any modality to be equivalent to a social object called information object.
- **Core Software Ontology (CSO):** Formalization of fundamental concepts in the computer domain, e.g., software or data.

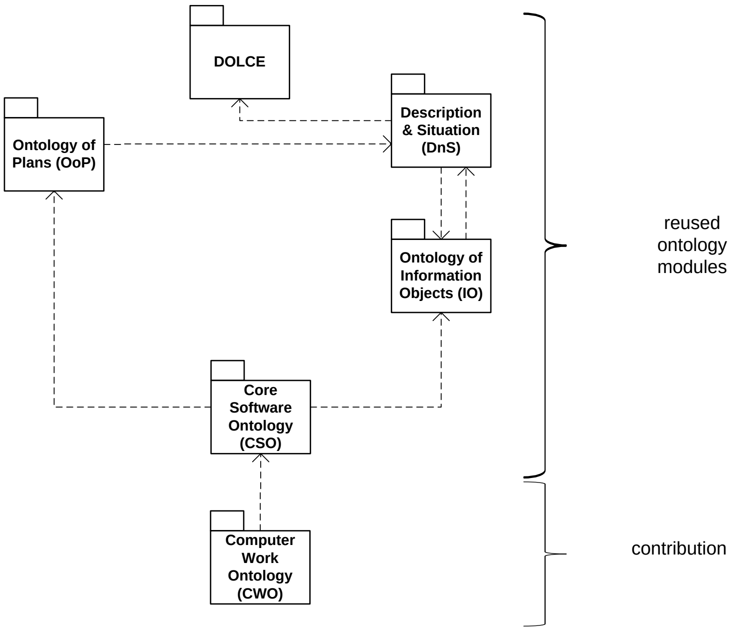

Figure 6.6.: Overview of the ontologies. Dotted lines represent dependencies between ontologies. An ontology O_1 depends on O_2 if it specializes concepts of O_2, has associations with domains and ranges to O_2 or reuses its axioms.

Following [200] four additional requirements need to be met to assure that an ontology is actually a specification of a conceptualization: The design of the ontology needs to (1) avoid conceptual ambiguity, (2) axiomatize existing concepts, (3) avoid concepts that have no ontological meaning but exist for modeling reasons only and (4) provide concepts without limiting future extensions of the ontology.

6.4.2 Computer Work Ontology: Computer Workplace Environment

The computer workplace is modeled as an environment that offers functionalities of generating, displaying and transforming data which can be consumed as information. The functionalities and the available information define a possibility space for the execution of work. Functionalities are encapsulated in software tools and information is stored in files.

6.4.2.1 Software and Functionalities

To model software, the respective design pattern as described in [200] is used: CSO:Software[3] is defined as CSO:Data that OIO:expresses an OoP:Plan, itself sequencing a set of OoP:Task (see Figure 6.7). The idea is to take a perspective on software as it is available to end users. The functionalities offered by the software are modelled as CWO:Functionality, a specialization of OoP:Task. To describe the plans, describing the purpose of use of a software (e.g., word processing), CWO:Scenario is modeled as specialization of OoP:Abstract-Plan. The CWO:Scenario sequences a set of CWO:Functionality.

(D1) CSO:Functionality(x) $=_{def}$ OoP:BagTask(x)
$\wedge\ \exists$ y (DOLCE:part-of(y,x) $\wedge$ ComputationalTask(y))

(D2) Scenario(x) $=_{def}$ OoP:Abstract-Plan(x)
$\wedge\ \forall$ y (DnS:defines(x,y) $\rightarrow$ Functionality(y))

(D3) CSO:Application(x) $=_{def}$ CSO:Software(x)
$\wedge\ \exists$ y (OIO:realizedBy(x,y) $\wedge$ CSO:ComputationalObjects(y)
$\wedge\ \forall$ z (OIO:expresses(x,z) $\rightarrow$ Scenario(z))

To give an example, the earlier discussed example of writing a report about project activities is picked up. To write the report, a template document was searched, using the Windows Explorer. This provides functionalities to open folders and display files, and is used to identify the template document.

(Ex1) CSO:Software(windowsExplorer)

[3] For all axioms and examples, entities that belong to CWO are given without prefix while all other entities are noted with prefix. To maintain the readability, all prefixes are given in the text.

(Ex2) Scenario(folderStructureInteraction)
(Ex3) Functionality(browseFolderStructure)
(Ex4) Functionality(getElementDetails)
(Ex5) Functionality(executeElementWithApplication)
(Ex6) OIO:express(windowsExplorer,folderStructureInteraction)
(Ex7) DnS:defines(folderStructureInteraction,browseFolderStructure)
(Ex8) DnS:defines(folderStructureInteraction,getElementDetails)
(Ex9) DnS:defines(folderStructureInteraction,
executeElementWithApplication)

6.4.2.2 Information Objects Represented by Files

Files realize a connection between meaningful information and software by data in a digital encoded representation. A CWO:File is a role played-by only CSO:Data. As CSO:Software is a subclass of CSO:Data, the played-by relation covers software as files (see Figure 6.7). CSO:AbstractData is another subclass of CSO:Data, containing data that identifies something different from itself, e.g., the word *tree* that stands for a mental image of a real tree. As a file may be abstract data or software, two aspects of files are supported: 1) being a static information object 2) being an information object for execution to make plans accessible in a runtime representation. A file as a static information object is modeled by relating the file as CSO:Data by DnS:about with a DnS:description. A file as an executable information object relates CSO:Software with OoP:Plan by the DnS:expresses relation.

A CWO:File is DnS:ordered-by a CWO:File-Format. A CWO:File with specific CWO:File-Formats can be input for CWO:Functionality. This connection organizes the file access by functionalities, which may range from opening the file to display content in a word processor to the interpretation of a web page by a web browser.

(D4) File-Format(x) $\rightarrow$ IO:Formal-System(x)
(D5) specializes(x,y) $\wedge$ File-Format(x) $\rightarrow$ File-Format(y)
(D6) uses(x,y) $\wedge$ File-Format(x) $\rightarrow$ File-Format(y)
(D7) File(x) $=_{def}$ DnS:Role(x) $\wedge$ $\exists$y(ordered-by(x,y) $\wedge$ File-Format(y))
$\wedge$ $\exists$z(played-by(z,x) $\wedge$ (AbstractData(z) $\vee$ Software(z)))
$\wedge$ $\forall$f(inputFor(x,f)$\rightarrow$Functionality(f))
$\wedge$ $\forall$g(outputFor(x,g $\rightarrow$Functionality(g))

In the following, two examples for using CSO:File are given. The first example is for a file as role played by CSO:Data that is not CSO:Software. This means the aspect of being an information object is of primary importance. For this purpose the template document is modeled which is identified in the previously mentioned example and its connection to a word processor is shown.

(Ex10) IO:Information-Object(project-template)
(Ex11) DnS:description(project)
(Ex12) DnS:about(project-template, project)
(Ex13) File(ProjectTemplate.docx)
(Ex14) DnS:played-by(project-template, ProjectTemplate.docx)
(Ex15) File-Format(docx)
(Ex16) DnS:ordered-by(ProjectTemplate.docx, docx)
(Ex17) CSO:Software(microsoftWord)
(Ex18) Scenario(textProcessing)
(Ex19) DnS:expresses(microsoftWord,textProcessing)
(Ex20) Functionality(openTextFile)
(Ex21) DnS:defines(textProcessing, openTextFile)
(Ex22) DnS:inputFor(openTextFile, ProjectTemplate.docx)

The second example is for a file as a role played-by CSO:software. This means that the file as software gives access to functionalities. Interesting examples are web applications interpreted from the perspective of a user. For a user, a web application is an address to be typed into a browser. By focusing on this aspect of consumption, the web application is software that plays the role of a file. In the following, the example of accessing a website which contains a project planning tool is provided.

(Ex23) CSO:Software(projectPlanner)
(Ex24) Scenario(accessProjectPlan)
(Ex25) DnS:expresses(projectPlanner,accessProjectPlan)
(Ex26) File(www.projectplanner.net)
(Ex27) File-Format(html4.0)
(Ex28) DnS:ordered-by(www.projectplanner.net, html4.0)
(Ex29) DOLCE:played-by(www.projectplanner.net, accessProjectPlan)
(Ex30) CSO:Software(firefox)

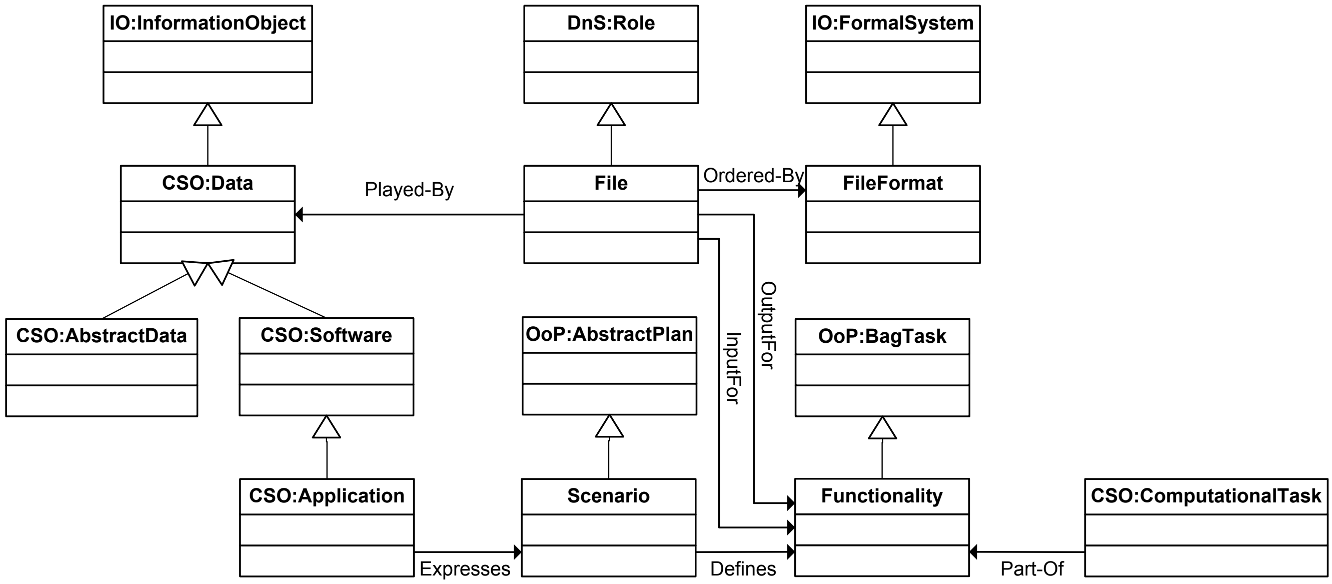

Figure 6.7.: The classification of software with scenarios, functionalities, and files. Concepts taken from DOLCE and accompanying ontologies are labelled with the respective name space.

(Ex31) Scenario(webBrowsing)
(Ex32) DnS:expresses(firefox,webBrowsing)
(Ex33) Functionality(accessWebsite)
(Ex34) DnS:defines(webBrowsing, accessWebsite)
(Ex35) DnS:inputFor(openWebsite, www.projectplanner.net)

6.4.3 Computer Work Ontology: Activity Execution

Work execution has been described as goal realization based on activities (see chapters 2, 3). Activities are executed by applying knowledge actions which are composed of desktop operations. The analysis has shown that knowledge actions are not executed sequentially, but that information workers frequently switch between different knowledge actions. As a result, one knowledge action is composed of a set of work episodes, each episode denoting the active work on the knowledge action. To model these episodes, application actions are introduced which are an additional structure between knowledge actions and activities.

The four elements are modeled as follows. Work is composed of Activities (CWO:Activity) which are composed of knowledge actions (CWO:KnowledgeAction). Knowledge actions are composed of application actions (CWO:ApplicationAction) while application actions contain desktop operations (CWO:DesktopOperation).

The connection of the elements is modeled using the plan pattern of the OoP [95] (see Figure 6.8). Modeling the task execution based on the OoP:AbstractPlan stresses the weak structure and adaptation of execution processes based on constraints. An OoP:AbstractPlan describes methods for the execution of a procedure. A CWO:Activity is internally represented in an agent, has a goal, and uses at least one CWO:KnowledgeAction. Following a hierarchical model, each CWO:KnowledgeAction uses a CWO:ApplicationAction, which uses a CWO:DesktopOperation. A CWO:KnowledgeAction references a description it is about. A CWO:ApplicationAction references a CWO:SoftwareClass, which organizes software that shares similarities with respect to the tackled scenarios.

(D8) DesktopWorker(x) $=_{def}$ DnS:rational-physical-object(x)
$\wedge\ \exists$ y(internally-represented-by(y,x) $\wedge$ Activity(y))
(D9) SoftwareClass(x) $=_{def}$ DnS:Collection(x)
$\wedge\ \forall$y(DnS:member(x,y)$\rightarrow$Software(y))
(D10) Activity(x) $=_{def}$ OoP:AbstractPlan(x)
$\wedge\ \forall$ y(ComplexTask(y) $\wedge$ uses(x,y) $\rightarrow$ KnowledgeAction(y))
(D11) KnowledgeAction(x) $=_{def}$ ComplexTask(x)
$\wedge\ \forall$ y(uses(x,y) $\rightarrow$ ApplicationAction(y))
$\wedge\ \exists$ z(references(x,z) $\wedge$ DnS:Description(z))

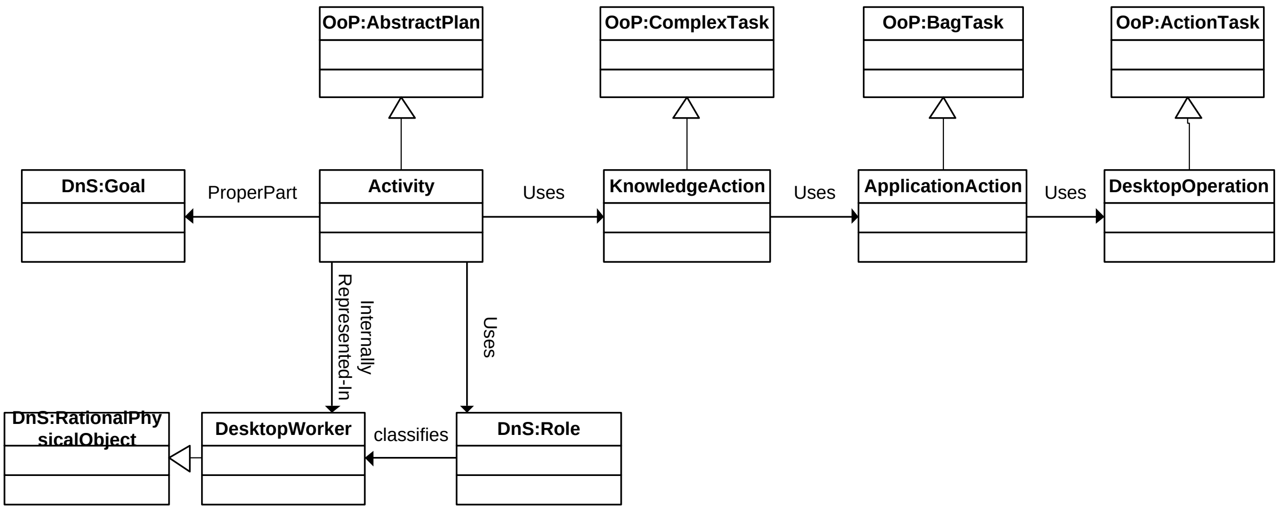

Figure 6.8.: The classification of the action hierarchy including Activity, KnowledgeAction, ApplicationAction, and DesktopOperation and use of the planning pattern. Concepts taken from DOLCE and accompanying ontologies are labelled with the respective name space.

(D12) ApplicationAction(x) $=_{def}$ BagTask(x)
$\wedge\ \forall$ y(uses(x,y) $\rightarrow$ DesktopOperation(y))
$\wedge\ \exists$ z(references(x,z) $\wedge$ CSO:SoftwareClass(z))

(D13) DesktopOperation(x) $=_{def}$ ActionTask(x)
$\wedge\ \exists$ y(uses(x,z) $\wedge$ Functionality(y))

To give an example,the browsing of information relevant to give a report on the status is given.

(Ex36) DesktopWorker(user)
(Ex37) Activity(reportProjectStatus)
(Ex38) internally-represented-by(reportProjectStatus,user)
(Ex39) KnowledgeAction(browse)
(Ex40) description(projectInformation)
(Ex41) references(browse, projectInformation)
(Ex42) ApplicationAction(searchFile)
(Ex43) defines(browse,searchFile)
(Ex44) SoftwareClass(fileBrowser)
(Ex45) references(searchFile, fileBrowser)
(Ex46) DesktopOperation(openFolder)
(Ex47) defines(searchFile,openFolder)

6.4.3.1 Object: Work Execution

Work in CWO is described as a hierarchy of knowledge actions, application actions and desktop operations sequenced by a plan.

In the CWO, tools are modeled as software expressing scenarios that define functionalities. The mediation by a tool includes a process of tool selection, as the subject identifies a tool that sufficiently supports a given goal. To model this mediation process, the CWO:sufficient-implementation relation is used as a specialization of DnS:intensionally-references. CWO:sufficient-implementation expresses that an OoP:task can be adequately executed by using a respective DOLCE:endurant. The CWO:sufficient-implementation is used to connect the CWO:KnowledgeAction and the CWO:DesktopOperation with software and functionality as tools to model the possible space of work execution (see Figure 6.9).

Although the mediation process is modeled, the actual execution of work is not represented in the ontology. Such a modeling would require the description of the actual perdurants carried out by the user, such as clicking with a mouse or typing with a keyboard (for an example, see [207]). Since the focus is rather on the abstract work processes themselves than their modality dependent execution, this level of detail is not included in the CWO.

(D14) KnowledgeAction(x) $=_{def}$ OoP:ComplexTask
$\wedge$ $\forall$y(sufficient-implementation(x,y)$\rightarrow$Scenario(y))

(D15) DesktopOperation(x) $=_{def}$ OoP:ActionTask
$\wedge$ $\forall$y(sufficient-implementation(x,y)$\rightarrow$ Functionality(y))

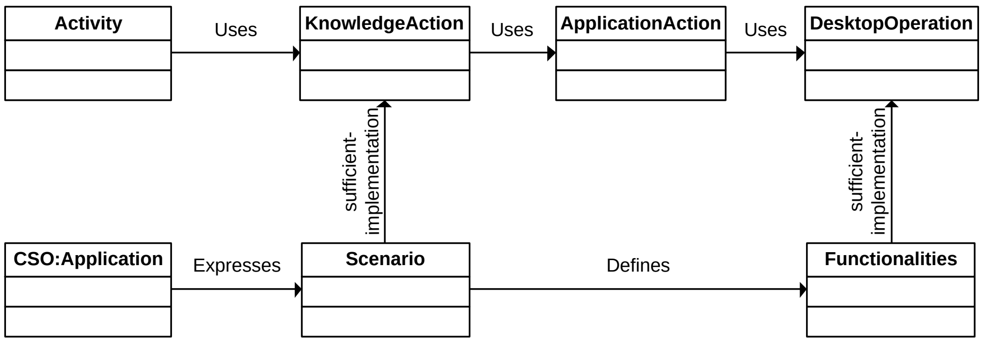

Figure 6.9.: The connection between the hierarchy of actions and software with scenarios and functionalities. Concepts taken from DOLCE and accompanying ontologies are labeled with the respective name space.

To illustrate the relation between activities, knowledge actions, application actions and desktop operations the relations are given for the example report project status activity.

(Ex48) Activity(reportProjectStatus)
(Ex49) KnowledgeAction(browse)
(Ex50) ApplicationAction(searchFile)
(Ex51) defines(browse,searchFile)
(Ex52) DesktopOperation(openFolder)
(Ex53) defines(searchFile,openFolder)
(Ex54) Software(windowsExplorer)
(Ex55) Scenario(folderStructureInteraction)
(Ex56) Functionality(browseFolderStructure)
(Ex57) OIO:express(windowsExplorer,folderStructureInteraction)
(Ex58) defines(folderStructureInteraction,browseFolderStructure)
(Ex59) sufficient-implementation(searchFile,folderStructureInteraction)
(Ex60) sufficient-implementation(openFolder,browseFolderStructure)

6.4.4 Intermediate Results

The described elements of the CWO formalize the information work process as it was identified within the course of this dissertation. By implementing the CWO into the ContAct monitor, a formal, reusable representation of activity execution at the computer workplace is provided. The description completes the data organization step for the ContAct monitor.

6.5 Related Work for ContAct Monitor and Computer Work Ontology

The ContAct monitor provides sensor based logging of interaction data, its processing and its organization. The output is a detailed description of the work process, using concepts like desktop operations and knowledge actions. In the following, related work is reviewed. First, monitoring applications are presented (see section 6.5.1). Second, formalizations in the context of monitoring and information work modeling are discussed (see section 6.5.2). Third, the outputs of other monitors are considered (see section 6.5.3). The section shows that a general focus on information retrieval for most monitoring applications results in a very narrow perspective of the work process—the work process as a collection of information objects. In contrast, the ContAct monitor provides a rich description of the work process.

6.5.1 Monitoring Applications

In the following different monitoring applications are reviewed[4]. In the following, the focus is interaction monitoring in the domain of information work support[5]. All considered applications use software sensors integrated into the operating system and into different applications, as described earlier for the ContAct monitor.

The structure and the organization of the monitors are given in table 6.2 and summarized in the following:

- **Architecture:** One can distinguish pull and push based monitors. A pull based architecture periodically checks for changed states. The pull frequency is an indicator of the degree of detail covered by the system. The most important advantage of a pull based system is the ability to create a direct connection between different context features. Whereas a push based system has to combine different events based on earlier cached information and needs to tackle problems of missing events, the pull based architecture actively requests all required information.

  Most reviewed monitors implement push based architectures. Only CAAD is a pull based monitor [217].

  The ContAct monitor is a hybrid application, using push and pull mechanisms. The contact monitor listens to events from different applications and especially to the mouse and keyboard stream. Once a mouse or keyboard event is fired, the system pulls related information. Advantages are the adaptation of the logged information to the actual work frequency of the user, while the events include all relevant information.

- **Covered applications:** The range of covered applications is a relevant indicator for the quality of the interaction history created by the monitor. If a monitor excludes different applications, it is not possible to derive the complete work process from the history. The reviewed monitors focus on different classes of applications. A separation which in some cases expresses different goals, in other cases expresses a preferred type of applications (e.g., preferring open source office software or Microsoft Office software). For those applications which do not offer an API, the collected information is very limited.

  The ContAct monitor covers the broadest range of applications in the set of considered monitors. The integration of the UI Automation framework ensures the collection of data even for applications which do not provide an API, as long as the considered application implements a Microsoft user interface framework (e.g., Windows Forms or WPF). The collection of content information is unique for the ContAct monitor. Swish [204] and CAAD [217] also use content data but for Swish it is limited to window titles, while CAAD only uses the MD5 (message digest algorithm 5) algorithm on the content to identify content changes.

- **Purpose:** Most reviewed applications have the purpose of addressing an information need identified based on the tracked interaction data. Only Swish is a research prototype, offering the data to an academic analysis framework [204].

[4] For a general and more detailed discussion of monitoring applications for interaction data collection, see section A.1 in the appendix

[5] Monitors for other purposes are not considered. For a discussion of monitoring with respect to user experience research, see [119]. Additionally, approaches which only consider monitoring data of one single application are not considered (for details on such monitors, see [298]).

	PETDL Monitor [16]	ICE library + X-Window hooks [54]	UICO [214]	TaskTracer [256]	APOSDLE [161]	CAAD [217]	Autohotkey based logger [32]	Swish[204]	Switch Logger [194]	ContAct [238]	CAM monitor [182, 298]	LIP monitor [235]	EPOS monitor [250]
Architecture	push	push	push		push and pull	pull	pull	push	push	push and pull	push	push	push
Keyboard			x		x					x			
Mouse			x		x					x			
Process			x		x	x				x	x	x	x
Active Window			x	x	x			x	x	x	x	x	x
File System			x	x	x	x				x			x
Clipboard			x	x	x					x			
Printer					x					x			
Applications:													
MS Word	x		x	x	x		x			x	?	?	
MS Excel			x	x			x			x	?	?	
MS PowerPoint			x	x	x		x			x	?	?	
MS Outlook	x		x		x		x			x	?	?	
MS Internet Explorer			x	x	x					x	?	?	
Mozilla Firefox	x		x							x	?	?	x
Mozilla Thunderbird			x								?	?	x
Google Chrome										x	?	?	
Other		WIMP supported	Novell GroupWise	MsVisual.Net			Adobe Reader						
Context Storage			RDF	?	relational DB	log files		relational DB	log Files	RDF-OWL	XML	Ontology	RDFS
	plugins required	Unix, Solaris and WIMP GUIs plugins required	MS windows XP/Vista events through app plugins	special events for Windows XP	special events for Windows XP	events for email, application state, and web browsing	only document path/ identifier	MS Windows window handle (HWND)	MS Windows window handle (HWND)				

Table 6.2.: Monitoring applications.

6.5.2 Formalization of Information Work

The ContAct monitor generates CWO ontology individuals based on sensor information. The resulting CWO ontology provides information about information work based on desktop operations and knowledge actions. In the following, related formalizations are mentioned which have the purpose of formalizing information work. A specific focus is given to formalizations which are integrated into the monitoring applications already considered (see table 6.2). Additionally, formalizations which are independent from the information work and focus on the information work place and information work processes are described.

6.5.2.1 Formalizations Integrated in Monitoring Applications

Various XML and ontology based formalization approaches for monitoring applications exist.

- **XML based formalizations:** The APOSDLE project has created an XML scheme for interaction events which classifies events based on the application, the type and the captured data (e.g., information object, receiver of an email, etc.) [161]. The APOSLDE event notion is used in the collection and processing phase of the ContAct monitor. Another XML based approach is contextualized attention metadata [182, 298]. Contextualized attention metadata (CAM) has been designed to collect interaction metadata for different persons as data groups. For each person a data group is created, storing the interaction data of different applications for the respective person. A specific aspect of attention metadata is the consideration of context information like a course or a situation the data was collected in.

- **Ontology based formalizations:** Different ontologies were proposed to capture the data. The LIP context ontology [235] structures personal, technical, social and organizational context features. Those features are extracted based on heuristics using monitored software sensors of user interactions. The heuristics are use case specific, e.g., “specific to the company environment” [235].

  Schwarz [250] created the Native Operations ontology (NOP) and the work context ontology. While the NOP ontology focuses on native operations on data objects, the work context ontology models attention which is calculated using NOP individuals. The work context ontology follows the idea of different context factors like “Personal, Social, Task, Environment, Spatial, Temporal, Informational”. The UICO ontology [215] is closely related to the work on NOP and the work context ontology. UICO captures operations like select, create, copy and maps them to objects like presentations, spread sheets etc.

Other formalization approaches focus on the manual modeling of task descriptions which are matched by monitored interaction events. Bailey et al. [16] use the Pattern-based Event and Task Description Language (PETDL). PETDL is an XML based language to describe tasks hierarchically, and to capture events which occur on application level. A similar approach of manual hierarchical modeling is given with [54]. Because those approaches do not generate the data structures automatically but require manual modeling for each individual, they are not considered further. Although potentially related, no information was given for the formalizations used in [194, 32, 204].

The described XML schemes and ontologies [182, 298, 161, 235, 215, 250] go beyond the mere mapping of sensor events to an ontology. The identification of context or attention is the main purpose of CAM, LIP, NOP and UICO to “support human attentional processes” [226] and identify an information need. Therefore, additional elements like the context element in CAM, the organizational and social context in LIP, an information need in UICO and a UserWorkContext and ContextualElementGrouping in the work context ontology exist.

6.5.2.2 Formalizations for the Workplace and the Work Process

Despite the extension beyond mere sensor information, there are limitations of the schemes and ontologies—having a monitoring focus—with respect to the degree of detail, information work at the computer workplace and the work process which is modeled:

- **Information workplace focus:** Social semantic desktops like Nepomuk [103] and IRIS (Integrate. Relate. Infer. Share.) [56] are information management systems which structure information work related data (e.g., files, emails, locations, tasks, etc.) in semantic networks. Both projects provide an initial ontology which allows a basic classification of things. The ontology of the Nepomuk project is a RDF-S ontology named PIMO (Personal Information Model). Similarly, IRIS provides a personal topic map based on OWL ontologies.

  The initial ontology is filled by crawling data stores. Additionally, it is assumed that the user begins to maintain and extend the ontology manually. The reviewed literature did not provide information about the experience with respect to the manual extension of the ontology. The ontology provided initially mainly focuses on the relation between information artifacts. An experimental integration of the work by Lokaiczyk et al. [161] and Schwarz [250] into the Nepomuk system exists. Such integrations offer an additional method of maintaining the information model and to use it for information retrieval tasks. A direction which is followed by the ContAct monitor with the CWO ontology with its use of the DOLCE ontology.

- **Process focus:** Formalizations of execution processes generally provide a vocabulary to specify goals and realize sequential or hierarchical task decomposition. The decomposition is realized by elements like *Object* and *Activity* in [105] or *Goal* and *Act* in the "Act Formalism" [192]. Modeling of the domain and knowledge-intensive planning are not tackled in depth by the reviewed approaches. Josephson et al. [51] underlines the relevance of using domain knowledge next to task knowledge—a direction explicitly followed by Brusilovsky's [38] work on plan and domain models.

  Execution processes in information work are considered by Catarci et al. [49]. Catarci et al. present a task specification language intended to be readable by machines and humans with a focus on planning and task decomposition. The task specification language considers work execution closely related to workflow languages like YAWL [280] and is inspired by hierarchical task analysis [8]. They envision the integration of interaction histories (they name them action logs) with the system to automatically extract task specifications. The preceding work [75, 48] focuses on form field entries, the logging of related information and the identification of information based on spreading activation algorithms.

  More formalizations of execution processes can be found in [95]. For most descriptions, the relevance of a workflow like description of the work process is striking. This is an important difference to CWO. CWO builds on the ontology of plans and uses it to create a process presentation which is guided by the combination of work techniques and not by the modeling of work process steps.

6.5.3 Monitor Application Output Formalization

The analysis of the monitors and formalizations of information work has shown that there is a focus on an information need in most monitoring applications. Logged information is used to derive a construct named context or attention and which is used to derive the information need. The approach of UICO [215] and the context ontology [250] already models specific operations which resemble desktop operations.

None of the reviewed monitors considers an equivalent to knowledge actions. Assuming that the interaction always represents an information need suggests that most monitoring applications consider interactions as application of information search techniques. Such a perspective is only useful if the only goal of an application is the support of one work technique. The goal considered in this thesis, information work support to decrease the frequency of memory failures, requires a broader consideration of information work. Therefore, the information workplace and the work process need to be formalized to cover more work techniques and the monitoring application needs capabilities to distinguish between different types of work. These aspects are given with the ContAct monitor by integrating monitoring functionalities with an in depth analysis of the information work process formalized in terms of the CWO. The use of DOLCE as the upper ontology ensures the extendibility and builds on the rich set of existing DOLCE extensions.

6.6 Summary

This section has presented the ContAct monitor and the CWO ontology. The ContAct monitor addresses the identified requirements RQ6 and RQ7 of unobtrusively collecting information about the user work process, its respective activities and involved elements.

The ContAct monitor realizes an interaction history management process consisting of a collection, processing and organization phase. The collection phase focuses on event streams which include content data from the user interaction. The processing phase involves the creation of desktop operations and knowledge actions, using rules for complex event processing to identify desktop operations and heuristics to identify knowledge actions within the collected data. The organization is realized using the CWO ontology which has been presented in detail.

The review of related work has shown that the ContAct monitor consequently continues existing work on monitoring applications. Existing experimental integrations of monitoring applications with rich ontologies like the personal information model are foundation for the CWO and the ContAct monitor. While most existing monitoring applications focus on the identification of an information need, the use of knowledge actions provides a broader presentation of the work context.

The proposed design addresses the identified privacy requirement (NF-R4): The ContAct monitor addresses privacy issues based on three aspects. 1) Based on the system architecture: The data collection, storage and processing are realized as local operations on the users' machine. 2) Data Transparency: To get an understanding of the data logged by the tool, the user can access the logged data. 3) User Control: If the user does not want to use a specific sensor (e.g., the keylogger), the respective sensor can be switched off.

7 Activity Mining for Information Work Based on Interaction Histories

This chapter investigates into activity mining for information work at the computer workplace. Here, activity mining refers to the discovery of information work activities based on an interaction history generated from information work. For information work, activity mining is a specific challenge due to the subject's freedom of action. Interactions which belong to different activities may be executed in rapid succession. To address this, activity mining needs to identify those interactions which belong to the same activity.

The idea of activity mining considered here, is the following. An activity mining approach has no information, except an interaction history (number and type of activities in unknown). Based on the interaction history sets of interaction clusters are identified which stand for activities. The purpose of activity mining for this dissertation stems from the identified requirements for a tool to address memory threats of information work execution (see section 5.4.5). The idea is to enable an increased work process awareness, improve the organization of work process related data and simplify activity switches. Therefore, information about activities, their connections, the related elements and the respective work processes is required. The crucial requirement underlying the mentioned requirements is the unobtrusive collection of the required data. The ContAct monitor presented in the previous chapter was the first step into that direction. Activity mining is the next logical step to address the requirements (RQ6-7).

An activity mining method should cluster those interactions of the user which have addressed the same goal. In this chapter three classes of activity mining methods are considered (see section 7.2). The classes have been designed to address the most striking features of activity data collected at the computer workplace: the work process (what is accessed when, how often and for how long?) and the textual content (what is the accessed content about?).

A method is considered as good if the identified activities include as many interactions which belong to the same goal and so few interactions which belong to different goals that an information worker accepts it as an activity representation. Thus, the subject's perspective on the activity influences the evaluation and it is necessary to consider this in an evaluation of an activity mining method. To address this problem, two evaluation setups have been created. First, an evaluation against a gold standard data set in a controlled study was conducted (see section 7.3). The controlled setup compares the performance among similar activities performed by different users. Second, interaction histories created during real information work execution at a company were used for activity mining. The people who executed the work had to assess the quality of the identified clusters (see section 7.4). Thus, the evaluations quantify method quality and indicate the usefulness for real information work execution. The chapter concludes with a discussion of the approaches (see section 7.5) and an overview of related work (see section 7.6).

Although, the chapter proposes and evaluates methods for activity mining, this is not the only idea of this chapter. The goal is to describe activity mining at the computer workplace as a specific type of problem to facilitate further research on the topic. Therefore, the problem is formalized first to provide a clear understanding of the problem characteristics and potential methods to address it (see section 7.1).

7.1 Activity Mining Problem for Information Work

This section formalizes the activity mining problem for information work at the computer workplace. The goal is to provide a better understanding of the activity mining problem in general which helps to better comprehend design decisions of the presented approaches.

The activity mining problem is shaped by the specific challenges of information work, especially multitasking and interruptions. Work is executed—using the terminology of this dissertation—by combining knowledge actions for different activities in work processes. Each knowledge action stands for a work technique applied on a resource with an application. Activities are collections of knowledge actions that serve a similar goal, in other words related knowledge actions. During information work unpredictable activity switches occur due to the interruption-based coordination of work (cf. section 3.2). As an effect the subject not only switches between knowledge actions which belong to the same activity but also between knowledge actions which belong to different activities. As an effect each knowledge action references interactions distributed over large time spans of the work process, interactions mixed with other interactions which—at least potentially—belong to other activities. In this respect, activity mining turns out to be a problem of knowledge action clustering. Those knowledge actions which belong to the same activity need to be in the same cluster despite the distribution and mixture of their respective interactions within the interaction history.

The distribution has an important effect on the transitions between knowledge actions: a knowledge action can be source or sink of more than two knowledge actions (i.e., the subject switched from/to more than one other application to the respective knowledge action). Therefore, interaction histories result in a graph of knowledge actions with weighted and directed edges, denoting the switches between knowledge actions (the respective representation can be generated by a tool, e.g., the ContAct monitor persists the data in the CWO ontology, cf. chapter 6).

With that said, the activity mining problem presents itself as follows (see Figure 7.1): Given is a graph $G = (V, E); E \subseteq [V^2]; V \cap E = 0$. The vertices V are knowledge actions. The edges E have weights $w : E \rightarrow \mathbb{N}_{>0}$ denoting the number of switches between the respective knowledge actions. Each knowledge action v has the properties type, content, duration and situations. Situations are a set of $(begin, duration, application, informationobject)$ 4-tuples. The situations are all application actions the knowledge action is composed of (cf. section 6.4.3), i.e., the period of time the knowledge action was active. The type is selected from the set of knowledge action types $K_{type} = \{ BROWSING, COMMUNICATING, AUTHORING, CONSUMING, ORGANIZING\}$. The duration $d = \mathbb{R}_{>0}$ accumulates all situation durations, thus the overall time the knowledge action was active. The content is a bag of words accessed or interacted with while the knowledge action was active. The content is composed of $(token_{id}, token_{count})$ $2-tuples$ for the number of times a word specified by the $token_{id}$ was used within the information object associated with the knowledge action.

Activity mining strives to identify a Clustering C as a set of $C_1, ..., C_n$ of non-empty disjoint sets such that their union equals V: $C = C_1, ..., C_n$ with $C_k = v_1, ..., v_n$ while $C_1 \cup C_2 \cup ... \cup C_n = emptyset$ and $C_1 \cap C_2 \cap ... \cap C_n = V$.

The assumption that the clusters are disjoint is a simplification. Observations within collected data sets (cf. chapter C) suggest that disjoint holds for most knowledge actions within a time segment of at least one week. Still, the stability of knowledge action to activity affiliations over time should be investigated further in future research.

Clustering quality derives from the purity and completeness of each cluster. Purity: a cluster needs to cover only one activity. Completeness: all knowledge actions which belong to the activity a cluster stands for need to be integrated. Purity and completeness can be judged best by the person who actually executed the work and produced the interaction history or by persons who are very familiar with the captured type of work[1].

The presented activity mining problem is designed to address two important goals (cf. the identified non-functional requirements in section 5.4). On the one hand, the user effort to get information about user activities is intended to be minimal. On the other hand, the background knowledge about the work process is assumed to be very limited (e.g., it is unknown what types of tasks may emerge). Next to these characteristics, two aspects of the knowledge action graph need to be emphasized. Aspects which are not directly visible based on the description but emerge when the graph is constructed based on real work execution data:

- **Process feature:** The graph models the work process based on the perspective of knowledge actions. Each weighted edge describes how often a user switched between two knowledge actions. Based on the interaction information it is also possible to identify the distribution density of the switches within a considered time segment. This data obviously carries information about the relatedness of knowledge actions.

- **Semantic feature:** The graph contains very large amounts of texts. Each knowledge action is composed of many low level events which capture textual content based on the user interaction and based on the extraction of text from accessed information objects. As a result, knowledge action nodes contain complete text documents (e.g., website content or emails).

The highlighted features will have relevance when it comes to addressing the activity mining problem.

7.2 Activity Mining Methods

This section introduces different activity mining methods which are evaluated in the remainder of this chapter. The goal is to provide a broad overview of methods applicable to design activity mining and to hint to promising directions. One of the presented methods will provide satisfying results which shows the overall feasibility of activity mining.

The directions of the approaches focus on the striking features of the knowledge action graph, the process feature and the semantic feature (cf. section 7.1). The literature review later in this chapter will show that these characteristics are obvious choices when it comes to activity mining. This is underpinned by some tests which were conducted in the course of this thesis to perform activity mining only based on information object urls or application types and which did not provide promising results.

The process and the semantic features are used to investigate into three research directions:

- Activity mining based on the semantic relatedness of knowledge actions (see section 7.2.1).

- Activity mining based on the process information encapsulated in the knowledge action node edges (see section 7.2.2).

- Activity mining based on a combination of semantic relatedness and process information (see section 7.2.3).

For each of the directions two different activity mining methods are proposed. The applied methods are well-known techniques from the domains of natural language processing (for the semantic feature) and graph clustering (for the process feature).

[1] This chapter is based on the paper [241]. Here, a more formal definition of the domain is given and browsing is filtered out. Although removing the browsing knowledge actions results in a slightly reduced F-measure for the gold standard, discussions with users have shown that the clusters are perceived as being more useful. The reason is that browsing knowledge actions frequently contain information which was used within the activity thus it belongs to the activity. However, if an information object is only considered during a browsing knowledge action it is of minor importance for the execution of the task, thus the amount of considered elements is increased without providing additional value.

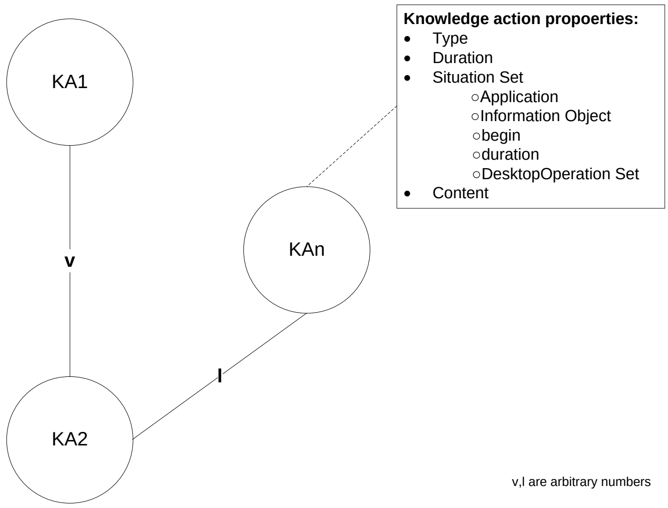

Figure 7.1.: Graph of connected knowledge actions. Each knowledge action is composed of a set of respective situations, type, duration and content.

7.2.1 Semantic Direction

The basic idea of the semantic direction is to cluster knowledge actions based on their semantic relatedness. This perspective assumes that activities always address certain topics and that the interactions related to the activity will relate to this topic based on the information they deal with.

Clustering based on semantic relatedness is realized based on a two step process. First, the semantic relatedness of the knowledge actions is identified. Second, the semantic relatedness is used as input for an agglomerative clustering. In the following, both steps are explained in detail.

7.2.1.1 Knowledge Action Semantic Relatedness

Semantic relatedness of knowledge actions is calculated based on the $k.content$ attribute of each knowledge action node. Basically, semantic relatedness refers to the degree to which two documents—in this case knowledge actions—address similar topics [269, 208]. Here, the goal is not to design new algorithms to calculate semantic relatedness but to investigate into the applicability of such algorithms for activity mining.

Three frequently used algorithms to calculate semantic relatedness are considered here. The algorithms have been chosen due to their acclaimed good performance on most data sets [27, 230]. All considered algorithms use a bag-of-words approach, i.e., a matrix that counts how often the words of the whole texts are used within each document is created [269].

To improve the quality of the calculations, a natural language processing pipeline was has been created [193, 89]. The goal of the pipeline is to extract only those words which carry much meaning, i.e., nouns and verbs. Therefore, the pipeline realizes a stopword filtering (remove words without semantic value, e.g., html tags), stemming (lead words back to their root, e.g., cars and car are lead back to car), part of speech tagging (identify the word type). As a result of the pipeline, only stemmed verbs and nouns were used to build the word count matrix.

The three algorithms considered here to calculate semantic relatedness are explained in the following:

- **Term Matching (TM):** The number of words that occur in both texts is calculated and scaled by the lengths of both texts (total number of words) [19]. Here, TM is considered as a baseline method.

- **Vector Space Model (VSM):** The Vector Space Model is an algebraic model to represent text documents [230]. Every text is represented as a term-TF*IDF vector (TF = term frequency, IDF = inverse document frequency) in the N-dimensional space (N representing the number of different terms in both documents). Text similarity is measured by the distance of the vectors within the model.

- **Latent Dirichlet Allocation:** Latent Dirichlet Allocation is a generative model which regards each text as a mixture of topics and traces each word's creation to one of the text's topics [27]. The model can be applied to realize topic detection and map each text to a probability distribution over the detected topics. The "distance" of two probability distributions can be obtained by utilizing a suitable divergence measure, e.g., Kullback-Leibler divergence.

The algorithms calculate a similarity value of two knowledge actions which can be normalized to a value between zero and one.

7.2.1.2 Clustering Based on Semantic Relatedness

The semantic relatedness is the input for an agglomerative hierarchical clustering algorithm [112]. Hierarchical clustering is an unsupervised algorithm which builds a hierarchy of clusters, given a set of input data, an appropriate distance metric, and a linkage criterion. The linkage criterion determines how the distance between clusters is computed. Here an average linkage clustering is used, which means that the distance between two clusters is computed by the average distance of all elements within one cluster to all elements within the other cluster. An agglomerative variant of hierarchical clustering is used, i.e., a "bottom up" approach. At the beginning, each knowledge action belongs to its own cluster. Then the algorithm finds the pair of clusters which has the highest similarity. Those clusters are merged into a new cluster and a new level of the hierarchy is created. The algorithm repeats this step until only one cluster remains and the hierarchy is complete, with respect to a threshold. Such thresholds need to be identified, e.g., using a calibration data set.

7.2.2 Process Direction

The second research direction uses the switches between knowledge actions as foundation for activity mining. This perspective assumes that the subject's movement between different applications and information objects indicates the activity membership, e.g., if a subject frequently switches between three knowledge actions it is assumed that they belong together (this builds on the insight gained during the analysis of information work processes, see section 3.3). Clustering nodes based on edges is generally referred to as graph clustering. Here, two algorithms for graph clustering are used to identify activities. Similar to the semantic direction, the idea is not to create new algorithms. The idea is to show the applicability of well-known algorithms to the problem of activity mining. Two approaches are presented in the following due to their popularity within different domains:

- **Markov Clustering Algorithm:** The Markov Cluster algorithm (MCL) [281] derives clusters from a graph based on a random walk, using Markov chains. MCL has been applied successfully in the domain of bioinformatics, e.g., to identify clusters within protein interaction graphs [286].

  MCL generates a transition matrix for a graph and simulates a random walk on the graph resulting in a step by step modification of the transition probabilities of the matrix. Knowledge action switch count is the basis for the initial transition matrix. Two processes are alternated for a given graph g and given parameters r (inflation parameter) and e (power parameter): expansion as taking the e^{th} power of the matrix and inflation as raising a single column to a non-negative power and then re-normalizing it. The idea is that strong neighbors are further strengthened while less relevant neighbors lose influence. In most cases the algorithm converges and allows the identification of clusters as non-negative values in rows. The resulting clusters are generally disjoint. The application of MCL to activity mining based on the transition matrix of the knowledge action graph is straight forward.

- **LinLog Approach:** The use of energy model based algorithms like LinLog for GraphClustering [197] has provided promising results in the domain of community mining [92]. LinLog produces a layout for graphs which considers the connectivity: Dense node connections result in spatial proximity while weak connections result in spatial distance. LinLog is a force based algorithm, i.e., a repulsive force is active among all nodes while connectedness generates an attractive force between nodes. Parameters for the LinLog algorithm are the forces a as the attractive force, r as the repulsive force.

  Using LinLog for the knowledge action graph will result in some knowledge actions being connected by edges longer than the average length, others will be connected by edges shorter than the average length. Edges which have a length above the average length are removed from the graph. The remaining components of the graph stand for activities.

7.2.3 Hybrid Direction

The previous directions have considered the semantic and the process feature one by one. In the following, the idea is that the combination of both might produce better results. To consider both features, they need to be combined by an approach. Two procedures are feasible. Either the semantic relatedness is translated into an edge and is considered by a graph clustering or the process influences the semantic relatedness. In the following, the process feature is used to influence the semantic relatedness. The influence is realized as follows:

- **Temporal gravity:** Semantic relatedness is weighted based on the temporal proximity of two knowledge actions. Time acts as a gravity force on the semantic relatedness. The technique follows the idea that temporal proximity influences semantic similarity: e.g., homonyms are understood based on a temporal proximity (asking someone for a bank during a finance conversation will be understood as a finance institute, not as the side of a river). The implemented process is as follows. The average distance between knowledge actions is calculated based on the shortest distances between the included interactions. The distance is the input to a function implementing the sigmoid function to calculate a value between 0 and 1: a long average distance between knowledge actions results in a value close to 0 while a short average distance results in a value close to 1. Weightings for appropriate interpretations of long and short need to be identified based on the considered data set.

- **Transition based cleanup:** The last step merges clusters based on the identified transition frequencies between the clusters. While the previous step only uses semantic similarity, this step considers the diffusion of two identified clusters. Thus, clusters that are not connected based on semantic similarity but which are connected by many switches are combined. If a user tended to switch very often between two clusters, i.e., used the applications and resources of both clusters simultaneously or in rapid succession, then the clusters are merged. For this purpose a distance matrix that counts all cluster switches is calculated. Clusters are merged based on a threshold value for the distance matrix. A minimal amount of switches needs to be identified and one cluster should not contain many elements. Otherwise, all clusters are merged step by step. The calibration of the respective values is a relevant step.

7.3 Gold Standard Evaluation

The previous section has presented different directions of activity mining research and respective methods. The goal of the next sections is to evaluate the cluster quality of those approaches. This section focuses on a quantification of the cluster quality based on a comparison with a gold standard. Such a comparison requires a controlled study setup and—while the quantification provides valuable insights—only simulates information work execution. To address this limitation, the following section complements the empirical analysis by evaluating real work data sets in another study.

Most of the described methods for activity mining require the specification of threshold or initialization parameters. A useful configuration needs to be identified before the methods are evaluated. Here, a calibration data set is used to identify an initial set of thresholds and parameters for the algorithms. The actual evaluation runs on a different data set. In the following, the two data sets—the calibration and the evaluation data set—are introduced and the evaluation process is described (see section 7.3.1). The evaluation of the activity mining methods follows. The evaluated methods are: semantic direction (LDA, VSM, Term Matching), process direction (MCL, LinLog) and the hybrid direction used with VSM as semantic similarity measure (see section 7.3.2). Additionally, an evaluation of the VSM algorithm executed only on the window titles is made. This investigates into the relevance of (computation intensive) content data in contrast to (not computation intensive) window titles.

7.3.1 Evaluation Process and Dataset

The evaluation uses two labeled interaction histories. One interaction history is used as calibration data set, the other one is used as evaluation data set. The purpose of the calibration data set is to identify optimal parameters or threshold values required for the different algorithms (e.g., a threshold for the agglomerative clustering, initialization parameters for the markov clustering algorithm). For the parameters identified with the calibration data set, the methods are evaluated with the evaluation data set.

Both data sets provide interaction histories for the execution of knowledge-intensive tasks. Nevertheless, they contain execution data from different users on different tasks. Due to the different tasks the use of a calibration data set and a different evaluation data set avoids an optimization for a few specific tasks. In the following, the data sets are only described roughly (see section 7.3.1). For details about the data sets (demographic information about participants, relevant tasks, etc.), see the appendix section C.1 and C.2)).

The two data sets have the following basic characteristics:

- **Calibration data set:** The data set "Data set1: Exploration data set – Controlled mono tasking work execution data" (see section C.1) has been used to identify parameters (e.g., clustering thresholds) for the activity mining approaches considered in this chapter. The data set contains 21 annotated interaction histories for the execution of 7 different knowledge-intensive tasks. To identify parameters, 21 interaction histories were analyzed. The overall interaction histories include 120 different task executions (not every task was executed by every participant due to time restrictions) and for each of the histories between four and seven clusters needed to be identified by activity mining. For the different approaches, different parameter configurations were iterated to identify the best configuration.

- **Evaluation data set:** The data set "Data set2: Gold standard data set – Controlled multitasking work execution data" (see section C.2) has been used to evaluate the approaches with the parameters derived from the reference data.

  The study included eight participants. The participants executed a set of predefined, knowledge-intensive tasks (see table 2). Five participants had post-doctoral positions and three participants were PhD students. The tasks were executed in random order and were disrupted during execution. Disruption means that tasks were interrupted randomly to generate activity switches as shown in Figure 7.2. During the execution of the tasks an interaction history was captured, using the sensor application. The created interaction histories were used as input to the activity mining method discussed in the previous section.

Participant	Knowledge actions to cluster
User 1	26
User 2	25
User 3	10
User 4	18
User 5	18
User 6	20
User 7	20
User 8	24

Table 7.1.: Number of knowledge actions per user in the gold standard set. Configuration: Browsing knowledge actions filtered out. Knowledge actions with minimal duration of 20 seconds.

– *Gold standard generation:* The study supervisors used activity data generated by the ContAct monitor and notes taken during study execution to validate the quality of knowledge actions and to create clusters of knowledge actions that were labeled with the respective task numbers. Thus, the gold standard assigns a task number to each knowledge action extracted from an interaction history (the knowledge action count per participant is given in table D.1).

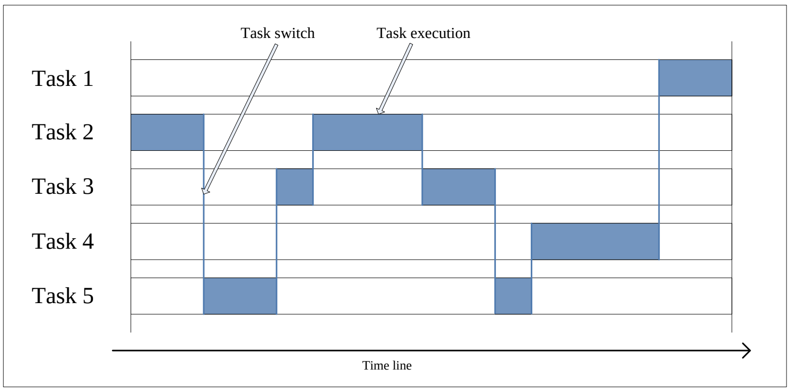

Figure 7.2.: Example for task execution process with activity switches.

The evaluation process is as follows. The reference data set is used to identify parameters for the approach. Second, the approach with the identified parameters creates clusters for the labeled gold standard data set. The mined clusters should be similar to the manually labeled clusters of the gold standard, i.e., it should be the same number of clusters containing the same knowledge actions. In order to compare the knowledge action clusters of the gold standard with the corresponding clusters identified by the system, the following labeling method was applied for each identified cluster: 1) Select a cluster from the activity mining as *tolabel* 2) Select the gold standard cluster with the largest percentage of knowledge actions matching the selected cluster as *compareCluster* 3) Label the *tolabel* cluster with the label of *compareCluster*. Three quantitative measures are extracted: 1) Precision: The fraction of knowledge actions in a mined cluster compared to the *compareCluster* of the gold standard in terms of true positives, false positives, false negatives, true negatives (for details, see section D). 2) Recall: The fraction of all knowledge actions in a manually labeled cluster corresponding to the *compareCluster* . 3) F-measure: The weighted harmonic mean of precision and recall ($F = 2 * \frac{precision * recall}{precision + recall}$). Generally, The higher the value of the F-measure the better the result of the algorithm [173].

7.3.2 Evaluation Results

The results of the gold standard evaluation indicate an overall feasibility of activity mining—at least on a controlled data set. Nevertheless, the results also show that some directions seem inappropriate to the problem. The results of the evaluation are provided in Table 7.2. For each method, the accuracy, the precision and the recall is reported. Additionally, an overview of the F-measure values of each approach for the different user data sets included in the gold standard is given in Figure 7.3.

The results provide insights in the applicability of the different directions and respective methods:

- **Semantic direction:** To evaluate the performance of different textual similarity measures, three different cluster distributions (TM, VSM, LDA) were produced and used as the input for the hierarchical clustering algorithm. The algorithm requires a

	LDA	TM	VSM	Hybrid VSM	MCL	LinLog	VSMWindowTitle
Accurracy	0.758	0.832	0.853	0.875	0.318	0.818	0.811
Precision	0.572	0.688	0.734	0.719	0.381	0.615	0.726
Recall	0.613	0.434	0.561	0.728	0.568	0.615	0.248
F-measure	0.59	0.53	0.63	0.72	0.46	0.615	0.37

Table 7.2.: Average results of the different task similarity measures.

threshold as a termination criterion. Thresholds for TM, VSM and LDA were identified based on the calibration data set resulting in three threshold values: VSM=0.15, LDA=0.9, TM=0.05.

The results for VSM (F-measure 0.63) clearly surpass those of both LDA (0.59) and TM (0.53).

The weak results of LDA are surprising. This result can be partly explained by the amount of input which is used to perform LDA. Only data collected from an independent run of the task detection system was used for the similarity calculation step. This is the setup for each similarity algorithm. With an increasing amount of data available, the quality of the inferred topic model most probably would increase. This should be the focus of further investigation regarding the applicability of topic models for task similarity.

The use of the window title to perform clusters provides an F-measure of 0.37. This shows that the amount of information required to deliver good results needs to exceed the data provided by the window title. In the current setup the clustering based on the VSM delivers the best results.

- **Process direction:** The two approaches perform very differently. While the overall F-measure of MCL is 0.46, the LinLog approach provides an overall F-measure of 0.62.

  A closer investigation of MCL shows that the result quality varies drastically between the different user data sets. A problem which was already visible when the parameters were identified on the calibration data set. An example within the gold standard data set: for user 1 the algorithm creates very many small clusters. Test modifications of the parameters only resulted in few very large clusters without improving the result.

  LinLog performs nearly as well as the VSM approach although it only uses switch information. This is a very interesting result as the collection of the data required for the process direction is much simpler than the collection of the textual content required for the semantic direction.

- **Hybrid direction:** The approach which combined VSM with temporal data to influence the similarity and to clean up the resulting clusters based on switches between the clusters performs best among all considered approaches. An overall F-measure of 0.72 is reached and only for two of the eight user data sets within the gold standard an F-measure below 0.6 was reached. A closer investigation into the algorithm has shown that the most important effect is gained by the cleanup step while the influence of gravity is limited (without gravity an F-measure of 0.69 is reached).

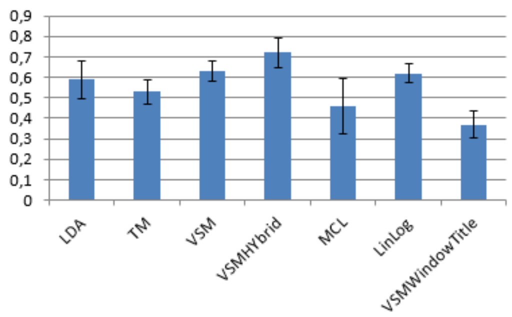

Figure 7.3.: F-measure and respective standard deviation for VSM, LDA, VSM with window titles only, VSM Hybrid, MCL, LinLog.

7.4 Work Data Based Evaluation

In the following, a complementary study is provided. While the first evaluation focused on a general quantification of the cluster quality of the different introduced methods, the second evaluation focuses on the subjective performance of the methods on real work data. The gold standard data set used “artificial” tasks to create comparable interaction histories for different users. The work data based study uses real work data sets collected during a longer period of work time. The subject who performed the monitored interaction evaluates the activities identified by a subset of the methods used for the gold standard analysis. Despite using all methods again, only those are considered which performed best in the previous evaluation, i.e., semantic similarity based on VSM, graph clustering based on LinLog and the hybrid approach.

To use real work data, a complex study process was required which is reported first (see section 7.4.1). The presentation of the results follows. This evaluation completes the overview of the research directions and their applicability intended with this chapter.

7.4.1 Evaluation Process and Dataset

The evaluation process is as follows. The ContAct monitor collects work data of a study participant for a period of five days with more than 50 % of computer work. The collected data is input for the activity mining approaches. Four approaches have been used. The VSM approach (semantic direction), LinLog (process direction) and the hybrid approach as the best performing approaches of the gold standard evaluation. Additionally, random clusters were used as a baseline.[2]

An application has been developed which randomly selects five single activity clusters from the data produced by several approaches for the user. For approaches generating less than five clusters the generated clusters are used. In a wizard mode the respective 25 (or less) clusters are presented to the participant. For each cluster the information object attached to the knowledge action contained in the cluster are presented. The participant ranks each cluster based on the following questions on a 7-point Likert scale from -3 to $+3$:

- I see one dominant activity represented by the displayed information objects.
- The collection of items I see looks random, there is no dominant activity.
- There is a dominant activity. Still, I see things that belong to other activities.
- I could continue working on the dominant activity based on the displayed information objects.
- When I see the information objects I remember the work I have performed on the displayed information objects.
- When I see the information objects, I remember that I still need to perform work related to the displayed information objects.

The questions focus on the purity of the clusters, their usefulness and their usefulness as memory cues. Aspects like cluster overlaps are not considered in this evaluation.

At the end of the study some questions related to the purpose of activity mining are asked:

- It is useful to see in a program on which activities I actually worked.
- It is useful to know how much time I spent on different activities.
- I would like to know how much time I spent procrastinating.
- I never search for information objects when I continue a task which was interrupted earlier.
- I am able to recall all activities I have finished during the last two weeks.

Six IT experts participated in the study. Some participants used the monitor for more than five work days. On each study day the participants did not perform less than 50 % of computer work. The participants were allowed to filter data from the data set before it was analyzed. Between the time of recording and the analysis was a time period of 5 to 10 days for all participants[3]. The parameters used for the approaches were similar to the parameters used in the gold standard analysis. While the participants used the application to enter their results, the study coordinator was present to answer questions.

Overall, 93 data sets from the activities mined based on the four used approaches were used within the study: 25 data sets for VSM, 18 data sets for LinLog, 25 data sets for Random and 25 data sets for the hybrid approach.

7.4.2 Evaluation Results

The results for the real work data sets are given in table 7.4. Most participants were skeptical about the approach when they participated in the study. One later said “I am surprised that some actually useful things were identified”. During study execution some participants were surprised to see information objects or complete clusters they obviously had forgotten to have been working on, indicated by statements like “I did that? ... oh yes, I did that. Already forgot that I checked this data.”

The results underline the usefulness of semantic similarity in the context of activity mining and show a limited usefulness of graph based clustering in its current design and configuration. The use of VSM shows the best results (mean 1.8, std 1.7) with only three clusters without any dominant activity. The participants report in most cases that they could continue working on the VSM clustered activities (mean 1.92, std 1.69). For most clusters no or few elements belonging to other activities were identified (20 votes for none

[2] Random clusters were generated with the following constraints: Between 6-12 clusters with 8-12 knowledge actions were created from the activity data.
[3] The Data set3: Activity mining data set small – Real world work execution data is described in more detail in the appendix, see section C.3

Type	Avg. cluster size
VSM	12.72
HYBRID	18.48
LINLOG	38.27
RANDOM	10.76

Table 7.3.: Average cluster sizes.

or few out of 25 votes). Participants are able to remember the work they performed based on the displayed information objects in most cases (retrospective memory, mean 1.28, std 2.09) and remember in eight cases that there is still work open, in four cases that the work was already done (prospective memory).

The hybrid approach is inferior to the VSM approach with respect to the cluster quality. Although dominant activities are perceived (mean 1.32, std 2.3) the result is lower than for VSM. For the hybrid approach as well as for the VSM approach one can say that those clusters which have a dominant activity are actually good clusters (in both cases most votes are given that none to few elements need to be removed from the clusters). The hybrid approach simplifies the act of remembering activities as well as the ability of resuming work to a higher extent than the VSM approach (mean 1.40, std 2.17 which is significant, T-stat=-2.4, $p<0.05$). A possible explanation for this: the hybrid approach creates larger clusters than VSM which have a lower purity while an overall information gain also with respect to the dominant activity of the cluster is achieved. The assumption is supported by the average cluster sizes, see table 7.3. While VSM clusters contain an average of 12 elements, the clusters of the hybrid approach have an average size of 18 elements.

The LinLog approach shows weak results for real work data. Many clusters contain information objects belonging to more than one activity (mean -0.5, std 2.06). Some are even perceived as random clusters (5 out of 18). The approach especially does not help people to remember work they conducted (mean 0, std 2.19). The main reason is the tendency of the algorithm to create very large clusters for real work data sets. This gives additional insight, as the approach worked very well for the gold standard (F-measure=0.62). Additional work on the parameters could improve the results.

As intended the RANDOM approach presents itself as baseline. This also indicates that the overall idea of dominant activities within clusters was understood by the participants.

At the end of the cluster evaluation different questions about the perceived usefulness of activity mining were asked (see table 7.5). The participants showed an overall appreciation of the idea of activity mining (Question1) and the gained insight into work (Question2). Nevertheless, many participants did not like the idea of knowing the amount of time they spent with information objects not related to work activities (Question3). The basic reasons for activity mining have been acknowledged by all participants: information searches are performed frequently after interruptions (Question4) and many participants were not able to recall all tasks they worked on during the last two weeks (Question5).

7.5 Discussion

- **Chances and limitations of semantic similarity:** For the gold standard an average precision of 0.72, a recall of 0.73, and an F-measure of 0.72 for the hybrid VSM approach was achieved. Some differences between the results for the different users are evident. For instance, the system was able to mine activities with an F-measure of 0.89 from the task execution of user 2. For user 4 the system achieved an F-measure of 0.52. One reason is the variety of accessed information objects. Especially websites tend to contain very many information types which are not directly related to the considered content (e.g., a flight booking website includes many advertisements and additional offers not related to the booking process of the user). The combination of semantic and temporal aspects was intended to mitigate this problem, but did not completely resolve it. The results for the real work data are promising as well, while the long term data shows that the likelihood of integrating too many useless knowledge actions in the clean-up step decreases the performance. Therefore, an approach completely relying on semantic relatedness performed better for the real work data.

- **Limitations of the process feature:** The good results of the LinLog approach for the gold standard show that a lot of information about the work process is included in the components of the knowledge action graph. Additional work on hybrid approaches which consider the semantic relatedness and the topology of the graph also seem to be a promising direction of future work on activity mining. Nevertheless, the robustness of the approach for real work data is not given in the current configuration.

- **General insights into activity mining:** The real work data information objects not belonging to work activities had specific influence on the evaluation results not covered by the gold standard analysis (as the gold standard did not include such elements). Only few participants filtered information objects from their activity data that was not work related. For most participants, accessed data included websites with private email accounts, social networks and news. In some cases, the use of these information objects is interwoven with certain activities. Thus—especially for approaches which consider the switches

Question1:	**I see one dominant activity represented by displayed information objects.**								
	-3	-2	-1	0	1	2	3	MEAN	STD
VSM	2	1	0	0	2	7	13	1,88	1.79
HYBRID	5	1	0	0	1	5	13	1.32	2.39
LINLOG	7	1	0	0	2	4	4	-0.05	2.59
RANDOM	12	2	0	1	7	3	0	-1.08	2.05
Question2:	**The collection of items I see looks random, there is no dominant activity.**								
	-3	-2	-1	0	1	2	3	MEAN	STD
VSM	15	5	3	0	0	1	1	-2.12	1.53
HYBRID	15	2	2	0	0	1	5	-1.36	2.43
LINLOG	5	3	2	1	0	2	5	-0.22	2.48
RANDOM	1	4	6	1	0	3	10	0.76	2.17
Question3:	**I could continue working on the dominant activity based on the displayed information objects.**								
	-3	-2	-1	0	1	2	3	MEAN	STD
VSM	1	0	1	4	2	3	12	1.92	1.69
HYBRID	0	1	1	0	1	7	10	2.48	1.41
LINLOG	1	0	1	0	3	5	1	2.22	1.90
RANDOM	3	0	2	2	3	2	2	1.84	2.44
Question4:	**There is a dominant activity. Still, I see things that belong to other activities.**								
	None	Few	Some	Many	NO ACT			MEAN	STD
VSM	13	7	2	0	3			-2.08	1.29
HYBRID	9	6	4	0	6			-1.48	1.55
LINLOG	5	1	4	0	8			-0.72	1.69
RANDOM	0	1	6	4	14			0.24	0.95
Question5:	**When I see the information objects I remember the work I have performed on the displayed information objects.**								
	-3	-2	-1	0	1	2	3	MEAN	STD
VSM	3	1	2	0	3	6	10	1.28	2.09
HYBRID	3	1	2	0	2	3	14	1.48	2.17
LINLOG	4	2	2	1	2	5	2	0	2.19
RANDOM	7	1	0	1	9	5	2	0.08	2.13
Question6:	**When I see the information objects, I remember that I still need to perform work related to the displayed information objects.**								
	-3	-2	-1	0	1	2	3	MEAN	STD
VSM	10	3	0	4	2	6	0	-0.88	2.08
HYBRID	10	2	2	3	5	2	1	-0.96	1.99
LINLOG	9	1	0	4	2	2	0	-1.27	1.91
RANDOM	13	2	2	3	4	1	0	-1.56	1.72

Table 7.4.: Results from the cluster evaluation for the different data sets.

	MEAN	**STD**
Question1: It is useful to see in a program on which activities I actually worked.	2.2	0.75
Question2: It is useful to know much time I spent on different activities.	2	1.09
Question3: I would like to know how much time I spent procrastinating.	-0.2	1.83
Question4: I never search for information objects when I continue an earlier interrupted activity.	-1.4	1.62
Question5: I am able to recall all activities I have finished during the last two weeks.	0.2	1.16

Table 7.5.: Average results of the different task similarity measures.

between the knowledge action—these not work related information objects were added to different activity clusters. The integration is reasonable if the switches indicate that reading a certain news item actually belongs to an activity execution process. Nevertheless, the study participant considered such information objects as unrelated, thus decreasing the cluster quality. A similar problem has been reported by [218]. A blacklist of certain domains would help to alleviate this problem.

Overall, the results from the real work study show a promising potential but they also indicate the relevance of further research in the domain. The high standard deviation for the cluster evaluation shows that the applied approaches tend to be on one extreme point: they create reasonable clusters or they create useless clusters. Approaches which consider different features of the knowledge action graph and are robust to data collected over long periods of time based on real work activities should be the focus of future research.

7.6 Related Work

Activity mining is applied in different research domains. In its most general form activity mining refers to the identification of performed activities from historic data. The historic data somehow encapsulates information about actions which happened in the world. Activity mining was performed based on news data [55], Twitter feeds [297], event logs of business systems [106] or user interactions [147].

In the following, different research which can be considered as activity mining for information workers is described and related to the approaches discussed here. The considered work is the CAAD system which mines clusters of resources which belong together [218], the Swish system which tries to mine activities based on window titles [204] and the work on routine task identification by Brdiczka et al. [34]. The focus is the identification of formerly unknown activities within interaction histories. Therefore, work on activity switch detection like [257] is not considered as long as the tasks which are the switch source and the target are not automatically detected.

The types of mined activities and evaluation methods as well as evaluation data sets substantially differ among the different approaches which deny a direct comparison. The CAAD system by Rattenbury et al. [218] is evaluated by a usefulness study. Activities are described in terms of activity/context units. Software sensors deliver information about the activation of those ACUs which is captured in a matrix. The identification of the activities is comparable to a modified non-negative matrix factorization.

Oliver et al. [204] report performance values for two data sets, one with 4 hours of user data. The process considered the window titles and the browsing direction which results in a graph visualization of the user work (although, the graph nodes contain less information than the knowledge action nodes described here).

Brdiczka et al. [34] perform activity mining based on document usage patterns identified by clustering events up to a threshold, i.e., the threshold value modifies the number of identified tasks. The activity mining is performed based on the sequential work execution process. An F-measure of 0.32 with a precision of 0.20 is reported for a data set of ten users and 50 tasks, collected over up to three work days. The results are explained by the amount of noise in the input data. By limiting the data set to the six most frequent tasks a F-measure of 0.74 is obtained.

From a model perspective, similarities between the considered approaches and the activity mining problem described here are obvious. Oliver and Rattenbury address a clustering problem which is performed on a graph or on a matrix. The important difference to the work described in this thesis is the type of abstraction gained with the knowledge actions and the relevance of content. Only Oliver considers the content and limits the use to the window titles. The study described in this chapter showed that a limitation of clustering to the window title decreases the result quality dramatically (cf. section 7.3).

Oliver et al. [204] report an F-measure of 0.58 for a data set of one user and five tasks, collected over approximately four hours. The results are improved to a recall of 0.76 % by using 1 hour chunks of data and application. A combination of probabilistic latent semantic indexing and a window switch matrix is used. Oliver et al. [204] is closely related to the approach presented here, as semantic similarity as well as a switching matrix is used. The difference is the amount of text used for the semantic similarity (cf. [204] limits the text to the window titles) and the task model (cf. [204] reports about the process but does not provide a task model).

The main difference to existing task mining approaches is the obtained task model. While the reviewed systems reduce activities to associated window titles [204], documents [34] or context structures (task relevant information and people) [218], the aforementioned activity mining method based on the activity data provided by the ContAct monitor identifies activities as clusters of connected knowledge actions, thus providing detailed information about the work process, its distribution over time, etc. In contrast to existing approaches, the mined activities can be used not only to enable support in the form of information object recommendation but process analysis becomes possible as well. To this point, this quality of support was only possible if an expert modeled all tasks manually [16, 54].

7.7 Summary

The introduction of activity mining at the computer workplace pursues two goals. On the one hand, activity mining is a basic requirement to provide information work support methods to address memory failures in information work. Therefore, it is necessary to specify activity mining as a basic technique to provide required data and to show the overall feasibility of activity mining. Next to this direct goal, the intention of this chapter has been to specify activity mining as a problem, show its relatedness to a well founded theoretical problems (clustering, natural language processing) and investigate into different possible research directions. This rather generic approach emphasizes the interesting research questions behind this problem and—hopefully—triggers further research in this domain. A request which is not only formulated in this dissertation. The need for activity mining for the domain of information work has been acknowledged by Shen in the conclusion on his thesis on activity recognition in desktop environments: “[...] in the future, I would like to explore how to aggregate resources into clusters without explicitly defining any task.” [254]

While pursuing the latter mentioned goal is not feasible within this thesis, the results indicate an overall feasibility of activity mining. For the three research directions (semantic feature, process feature, hybrid approach), a set of activity mining methods was introduced (see section 7.1). In all cases, the methods use well-known algorithms from the domains of natural language processing and clustering to realize activity mining. Two complementary evaluations have been conducted. A gold standard evaluation quantified the quality of the methods and provided an insight into the performance of the algorithms for real work execution data (see section 7.3). The best results for the gold standard were achieved with a hybrid approach, combining semantic similarity and graph topology (accuracy: 0.875, precision: 0.719, recall: 0.728, F-measure: 0.72). For the real work data the purity of the clusters was clearly increased for a pure VSM approach (one dominant activity: 1.88 on a 7-point Likert scale from -3 to $+3$, objects are contained that belong to other activities: -2.08 on a 7-point Likert scale from -3 to $+3$) while the recall of work and the perceived ability to continue work increased for the hybrid approach (recall activity: 1.48 on a 7-point Likert scale from -3 to $+3$, continue work: 2.48 on a 7-point Likert scale from -3 to $+3$). For details on the reported values, see Table 7.4. The graph topology approach using LinLog showed interesting results for the gold standard but did not perform well for the real work data.

The results show the general difficulty of identifying borders for clusters, as some clusters were too broad, others too limited. Another difficulty which became obvious based on the real work evaluation is the personal perception of activities. The use of not work goal related information objects (e.g., news websites) has become a fundamental element of the execution of some activities. The containment of those resources within activities is a correct result from an activity point of view. Nevertheless, the study participants did not accept the integration as they did not see the connection of those objects to their activity anymore. This is a problem of self-perception and opens the crucial question if a “boring excel fill out task” can only be completed if the news are read in between—interestingly enough, the literature on interruption indicates that there is a real positive effect of this type of activity (cf. section 3.2.4). These and other insights are relevant for further investigations into the topic and have been discussed in detail in section 7.5.

A review of related work has completed the chapter and provided indication that the proposed methods continue existing work on activity mining and that the achieved results are competitive (see section 7.6).

The overall feasibility of activity mining is the basic conclusion of this chapter. Having that said, information work support methods can be developed which use activity data to realize user support and decrease the threat of memory failures. The following chapter will investigate into such methods and will show that a decrease of memory failures based on support methods that apply activity data is feasible.

8 Information Work Support Methods: Design Space

This chapter specifies a design space for the development of support methods to address memory failures that occur during information work at the computer workplace. The design space is a collection of guidelines which have been created based on the functional and non-functional requirements for an information work support tool (see section 5.4). Based on the design space very different support methods can be developed. There is no single solution to address the requirements but different decisions can be taken during a design process. For this dissertation, the specification of a design space has two basic goals. First, the design space is constructed to facilitate method design and is used to design support methods which are presented in the next chapter. Second, the design space is a framework to assess support methods by tracing them back to basic design directions. The framework will help to compare the benefits and disadvantages of the support methods provided in the next chapter.

The design space builds on the activity data foundation specified in the previous chapters: networks of knowledge actions based on the processing of interaction histories (see chapter 6) and activity collections based on activity mining (see chapter 7). Two groups of guidelines are contained in the design space:

- **Design guidelines:** The first group of guidelines supports method design (see section 8.1). The guidelines address a basic challenge of memory support: once a memory failure occurs the information need is not known precisely—forgetting obviously means to not know exactly what was forgotten. To address this, the design space specifies three design directions. The design directions create different types of memory cues based on information related to the information need. A memory cue is an information which helps to remember a specific information. The next section derives these aspects based on an analysis of the functional requirements.

- **Interaction design guidelines:** The second group of guidelines supports interaction design for the support methods (see section 8.2). Guidelines for the interaction with activity data are provided. The interaction guidelines are based on the non-functional guidelines for a support tool. An important challenge is the design of visualizations which are appropriate for activity data. Respective research with an evaluation is reported.

The chapter finishes with a summary of the guidelines of the design space (see section 8.3).

8.1 Design Space I: Support Method Design

This section specifies design guidelines for user support methods based on activity data. The requirements RQ1-RQ5 need to be addressed by a user support method (see section 5.4 for the requirement specification). Unfortunately, it is not possible to derive support methods directly from the requirements. The requirements merely formulate information needs which occur during information work. Still, there is uncertainty *which* information needs will occur *when*. As a matter of fact the subject will never exactly know what was forgotten and thus will not be able to specify information needs explicitly.

The design space fosters methods which support recall processes. Despite focusing on methods which guess what information to deliver, the design space fosters method design directions which follow a simple idea: offer interaction with activity data which actively supports the recall process (see section 8.1.1). Overall, the decision towards an active support of recall processes to address the functional requirements results in the specification of three design directions (see section 8.1.2):

- Explorative methods—the subject explores data to address information needs.
- Organizational methods—the subject organizes data of high relevance.
- Recommender methods—the subject's awareness of earlier work is supported by recommending activities.

Based on these elements, this section shows the basic design ideas conveyed by the design space.

8.1.1 Basic Principle: Mediate Memory Cue Creation

The design space facilitates the design of user support methods which address this challenge by a basic principle: *Support methods should mediate the creation of memory cues based on activity data* (P1). This principle derives from the fact that the requirements formulate information needs which are addressed by complex recall processes. It is not possible to address the requirements directly but it is necessary to support the whole recall process. This is assured by the principle P1.

Data type	Activities	Knowledge actions
Properties	• duration • knowledge actions	• duration • application • information object • durations of included segments • relation to other knowledge action

Table 8.1.: Activity data properties

The basic principle builds on the need to address the identified functional requirements by support methods. Support methods should enable a subject to 1) derive activities (RQ1) 2) derive activity related elements (RQ2) 3) derive relations between activities (RQ3) 4) derive work execution processes and should additionally be supported by activity switches (RQ5) (see section 5.4 for the requirement specification). All requirements specify information needs to be addressed by a support tool. Nevertheless, it is unknown in which situations which information needs occur. It is possible to quantify the amount of memory failures and to identify the type of information which would address those failures. Still, it is not possible to foresee which information needs need to be addressed. Notably, talking about forgetting means that even the subject does not exactly know which information is missing. Therefore, it is not possible to address the requirements directly.

To address the requirements, a primary challenge is the investigation of the information need. To achieve this, the recall process of the subject needs to be supported. Human memory is associative, i.e., information is always encoded in relation to other encoded information [53]. Recall success heavily depends on the encoding specificity [277], i.e., the situation and the context of the encoding. For information work this specificity largely varies due to the interruption-driven work execution. Memory cues generally support recall by activating associations probably related to the desired information. For information work, memory cues are intended to support the recall of work related data, thus decreasing memory failure likelihood. Thus, support methods need to help a subject to create memory cues.

By interacting with activity data—comparable to an information need specification—the subject creates a memory cue, a stimulus for an automatic access or a subjective recall of required information (this follows from [277]). In this sense, the support methods support the recall process based on externalized information—which is the basic principle of the design space.

8.1.2 Design Directions: Exploration, Organization, Recommendation

The functional requirements have been used to identify tasks to be addressed by support methods and to identify a general approach of recall support based on memory cues. The memory cues are generated based on activity data. Activity data contains various information about the work process (see Table 8.1). The main challenge for a support method is to offer this data to the user in such a way that a recall process is supported. To address this, three design directions are introduced in the following (a decision for three directions which is informed by the structure of support methods described in the state of the art review, see section 5.3):

- **Exploration:** The subject explores activity data. The exploration process extends the recall process of the subject and combines the specification of the information need and the identification of the needed information.
- **Organization:** The subject actively organizes the data which is relevant. Organization means that the subject manually creates memory cues which become relevant later in the work process.
- **Recommendation:** The system recommends fragments of activity data to the subject. The recommender approach tries to guess which information is helpful in the work process and provides this to the user.

Memory cues can only be understood in reference to the associations of the subject. Some fragments are known, others are unknown. The different research directions address this problem by providing different ways of creating and consuming memory cues to identify and address information needs.

8.1.3 Design Foundation: Activity Data Interaction

A foundation for each support method is the structure of activity data. All methods are intended to mediate the recall process based on externalized activity data, no matter which design direction is chosen. To achieve this, the support methods need to offer modes of interaction to extract information from the activity data.

Activity data includes different properties which externalize relevant information about the work process (see Table 8.1). Support methods apply the properties to mediate memory cue creation.

The subject extracts information from the provided and encoded (e.g., visualized) activity data and integrates this interaction into the recall process. Therefore, support methods need to offer modes of interaction which facilitate the subject's information extraction tasks. To give an example, the subject searches for a document and remembers that he spent much time with it. Therefore, the subject will search for duration details within the activity data.

In the following, information extraction tasks are defined which cover the activity data properties. Two task classes are distinguished, tasks with the goal of improving the work process awareness to build up memory cues and tasks with the goal of retrieving information objects.

8.1.3.1 Information Need: Improved Work Process Awareness

The creation of memory cues is assumed to benefit from an overall work process awareness. The following tasks focus on the extraction of information to improve the work process awareness:

- **UD task (identify Usage Duration)**: The time spent with an activity is difficult to identify for information workers [24]. However, an improved temporal understanding of work is useful for planning future work as well as for time reporting (e.g., agile software development methods like SCRUM require an estimation of time investment. Based on activity data the estimations will improve).
  - ***Answers to:*** "I as an information worker want to know how much time was spent with an information object?"
  - ***Example:*** A user worked on a contract proposal named "contract.docx" for a couple of days. To settle the proposal creation, he/she needs to identify the time required to create the contract. He/she uses the actual working time with the document as hint to the total time needed.
  - ***Referred to-as:*** UD information need
- **UT task (identify Usage Time)**: This class involves information needs regarding the activity performed at a certain time, which activities followed other activities, or a sanity check whether a certain activity was actually performed.

  Due to the number of accessed information objects, and complex planning and replanning processes that are involved in information work, individuals forget aspects of their work process [239]. An improved structural understanding of work helps to avoid retrospective and prospective memory failures. Retrospective memory addresses remembering what was done. Prospective memory addresses remembering what was planned to be done. Especially in the context of interruptions both memory types are crucial, as failures of both types result in higher failure rates once work processes are intended to be resumed after interruptions [65].
  - ***Answers to:*** "I as an information worker want to know what was done in the beginning of the work process?" "What was continuously relevant?" "How did the work process proceed?"
  - ***Example:*** A user is asked to tell a colleague how he/she created a document. He/she needs to remember what he/she initially did and what he/she did after that and so on. Therefore, he/she needs to get on overview of the work process.
  - ***Referred to-as:*** UT tasks

8.1.3.2 Information Need: Associative Retrieval

Object retrieval is a relevant type of information to be extracted from activity data in recall processes. Due to the structure of activity data (textual content and associations of elements) two object extraction tasks are relevant:

- **DC task (retrieve by DesCription)**: Keyword based search for an information object included in the activity data.
  - ***Answers to:*** "I as an information worker want to know where I can find the document *doc* that I accessed earlier?", "I am looking for the document that is described by keywords a,b, which I accessed earlier..."
  - ***Example:*** The user searches an information object "sales report.docx" based on descriptive keywords.
  - ***Referred to-as:*** DC tasks
- **RO task (find by RelatiOn)** The RO process is an information object search based on the usage context of an information object. A user remembers certain objects he interacted with or a time segment and wants to identify the related resources. This especially addresses a context based memory.
  - ***Answers to:*** "I as an information worker search for an information object I accessed earlier, but I do not know enough to find it by description. But I remember other activities I executed at the same time."
  - ***Example:*** A user remembers that there was an interesting document, but does not know enough to enter a description; however, he/she remembers another document accessed while he/she wrote the first document. He/she searches for the remembered document and identifies the related document.
  - ***Referred to-as:*** RO tasks

8.1.4 Intermediate Results

This section has specified the design space for information work support based on activity data. The design space provides guidelines for support method design which have been derived from the functional requirements for an information work support tool (see section 4.5.2):

- **Basic principle:** Support methods need to mediate the creation of memory cues based on activity data. Memory cues facilitate a subject's recall process. If information is forgotten, the subject is not able to specify the information completely—which is the nature of forgetting. A recall process includes a specification of the known facts about the missing information. By specifying the information need it becomes possible to address the need.

- **Design directions:** Three different design directions have been specified, namely exploration, organization and recommendation. The directions address different requirements. Therefore, the selection of a design direction supports some types of recall support while others are neglected.

- **Design foundation:** The design foundation is activity data. Information extraction tasks have been defined which stand for the encoding of basic information required in the recall process. Support methods need to be designed to support a subset of the defined tasks to mediate recall processes. The actual implementation of the tasks highly depends on the chosen design direction. Nevertheless, each design direction needs to address the described tasks.

The three guidelines specify the basic decisions and necessities involved in support method design. The next section delivers interaction design guidelines for support methods. The interaction design guidelines are derived from the non-functional requirements for an information work support tool (see section 4.5.2).

8.2 Design Space II: Support Method Interaction Design

This section delivers interaction design guidelines for the support method design space. The goal of the design guidelines provided in the previous section is to specify a direction of information work support. This section specifies the way the interaction with the user should be designed to assure ease of use for the support method.

The interaction guidelines are derived from the non-functional requirements of the support tool (see section 8.2.1). Specific attention is given to the design of interactive visualizations. The visualizations need to address information extraction tasks specified in the previous section (see section 8.1.3). In this respect, the benefit of a visualization depends on the mental effort required to decode information required to address the information need, the visualization was developed for. Existing and new activity data visualization concepts are described (see section 8.2.3, 8.2.3) and discussed with respect to their appropriateness for the extraction tasks based on visualization theory (see section 8.2.2). The performance of the different visualizations for the required tasks is assessed by a user evaluation (see section 8.2.5)[1].

8.2.1 Basic Design Guidelines

The non functional requirements foster the design of an application which is simple to use and which meets user expectations. The design space is constrained by interaction design directions which have been identified based on the non functional requirements (see section 4.5.2)[2]:

- ***NF-RQ1: The use of the system should be simple, easy to learn:*** To simplify the interaction with the system, method design should follow Shneiderman's Mantra "overview first, zoom and filter, then details-on-demand" [259]. This can be assured by using in-place editing and expandable user interface elements in a comprehensible and similar way within the application.

- ***NF-RQ2: Permanent and simple access of the system:*** The support methods are relevant during normal work processes. Therefore, the simple access of the methods is important. Shortcuts and features like the task list jump bar should be used to provide quick and simple access to the methods and their functionalities.

- ***NF-RQ3: Operate efficient:*** The collection and processing of activity data is an operation intensive activity which threats the overall performance of the computer it is used on. The use of threads, caching and bundling of database calls should be considered to limit the resource consumption.

- ***NF-RQ5: The system should be accepted by the user:*** Technology acceptance is crucial for information work support tools. Therefore, user interface concepts and methods should be used users are familiar with.

1 The section is based on the conference paper [236].
2 The non functional requirement NF-RQ4 is not included in the list. The reason is that the requirement addresses privacy and no interaction related topic.

The non functional requirements identified in the first iteration focus on the simple use of the system. The requirements identified later consider additional aspects of the user: the technology acceptance and the effort involved in decoding required information.

8.2.2 Interactive Activity Data Visualizations: Characteristics

The remainder of this section focuses on guidelines for the design of interactive visualizations of activity data to address information extraction tasks. The topic results from the requirement *NF-RQ6: Information extraction from the system should be simple*. The work on support methods has shown that the topic is a critical success factor for the usefulness of a support method (the design cycle applied to develop the methods clearly shows this, see section 9.4). Graph based visualizations of activity data are in the focus of the investigation.

To solve an information extraction task, a subject needs to access the required information in a visualization and needs to decode (understand) it. Although decoding is an individual process, it follows certain regularities. In the following, general rules of visualization comprehension are discussed. The gained insight is applied to the previously identified information extraction tasks, to derive interaction design guidelines.

8.2.2.1 Human Information Visualization Decoding

Activity data visualization in a graph needs to consider how visualizations are decoded by humans. Reading graphs differs from the linear reading of text. The spatial organization of elements like basic shapes, texts and visual elements like images is used to transfer information. Good graph visualizations are designed based on this rich collection of elements to realize quick and successful information extraction by a user [295]. Limiting the mental effort required to decode required information from a graph visualization is a major challenge for graph design. Important aspects of visualizations are summarized in the following. The work builds on a theory of graph understanding by Pinker that connects graph understanding and involved mental effort [211].

Graphs communicate n-tuples of values on organized scales. Scales and values are encoded as visual objects that apply visual features to display information (length, position, lightness, shape, etc.). Graph understanding requires 1) an encoding of the physical dimensions of graphical elements and 2) an understanding of the meaning of the scales, the elements and the objects they stand for. The interplay of both aspects is crucial. A complex visualization requires high mental effort to decode the image and to identify the scales and the relation of objects to the scales. Objects that represent scales may realize a coordinate system. Based on the coordinate system, other elements are perceived and compared.

An easily consumable visualization is understood almost effortless. To realize this, laws of perception need to be applied to optimize the graph drawing with respect to the visualization goals. Important perception laws are formulated by Gestalt theory: proximity, similarity, common fate, good continuation, closure, figures, ground and connectedness [110]. The laws hint to those graphical formations that are decoded almost effortless by an individual.

Pinker [211] stresses that first the spatial organization of objects (following Gestalt theory), and then trained attributes are decoded following a decoding likelihood: the unconscious decoding of spatial organization reveals objects which are decomposed into scales and values. Values are directly decoded as being relative to the scales, and as being relative to all existing values. Only in a second step, conscious processes can enhance the understanding of the graph, requiring, however, mental effort and time. Different limiting factors complicate graph understanding. Individual processing capacity is limited. Human beings can separate between four and nine elements at a time. The number of elements is even less if processing resources are devoted to a concurrent task.

8.2.2.2 Requirements for Useful Activity Data Visualizations

Pinker's work [211] gives guidance on which aspects of graph design need to be considered to optimize the mental effort to decode information required to solve a specific task from a visualization. The visualizations are structured based on the information work support tasks which have been identified based on the functional requirements (see section 8.1.3).

To optimize activity data visualization for the identified information extraction tasks, certain requirements must be met. The following list contains visualization requirements that stem partly from the general principles for useful visualizations as discussed above, partly from several personal discussions about comprehension of activity data. Discussion which took place on the one hand with experts in information visualization, on the other hand with users of existing tools that integrate activity data visualizations (the visualization requirements "V-RQ" are a complimentary third requirement type next to the functional "RQ" and non-functional requirements "NF-RQ" for an information work support specified in section 4.5.2). The term simple encoding refers to the application of Gestalt laws to simplify encoding for the specific information type:

- ***RO-Tasks*** requirements:

  (V-RQ1) Simple encoding of relations: Relations between activity data elements should be easily identifiable, i.e., when the user has switched from one information object to another, this needs to be clearly visible.

(V-RQ2) Weighted relations: Relations should be weighted to display their relevance, i.e., when a user has switched frequently between two information objects, the frequency should be visible in the visualization.

(V-RQ3) Simple encoding of time segments: Time should be decomposed into discrete time periods, so called time segments, to structure activity data. Thus, a user can identify a certain time period like "yesterday morning" and see which activities were performed in that period.

- ***UD-Tasks*** requirements:

  (V-RQ3) also applies.

  (V-RQ4) Simple comparability of time data: Temporal data should be associated to a scale that enables easy identification of time segments, i.e., a user should be able to extract information like "happened before", "happened after" or "'happened while" easily.

  (V-RQ5) Preservation of process information: Activity data element presentation should show how the visualized work process was structured, so that a user can easily assess that information object *A* was accessed in the beginning, whereas information object *B* was accessed towards the end of the time segment under consideration.

- ***UT-Tasks*** Requirements:

  (V-RQ6) Simple encoding of usage times: The overall time the user accessed an information object in a specific time segment should be easily decodable. Thus, the user can easily see how much time information object *A* was accessed between e.g., 4PM and 5PM yesterday.

  (V-RQ7) Simple comparability of usage times: Usage time should be associated to object scales (following Gestalt laws) to enable a simple identification of values and direct comparison of the respective values. This way, a user can easily compare different duration times to extract "longer than" or "shorter than" information.

- ***General*** Requirements:

  (V-RQ8) Limit amount of perceptual units: The visualizations need to be understandable for large sets of activity data. As human perception capability is limited, the visualization needs to find useful ways to structure large data sets.

  (V-RQ9) Easy to learn: The visualization should not require extensive learning effort.

In the following, existing and newly developed activity data visualizations are assessed based on the specified visualization requirements.

8.2.3 Interactive Activity Data Visualizations: Existing Types

Different activity data visualizations are currently used both in commercial applications and in research prototypes to solve the discussed tasks. These tools use lists, line- and bar charts, Gantt charts, or grouped object sets (as proposed by Rattenbury [217]). In the following, the visualization requirements on bar-, line- and Gantt charts are focused.

Lists and grouped object sets are not discussed in more detail, as the former suffer from the large cognitive effort required to decode information as the list grows, and the latter focus on DC tasks and do not encode time or relation information beyond grouping objects according to a shared context.

8.2.3.1 Bar- and Line Charts

Bar- and line charts are the dominant visualizations for activity data-based analytics. These charts use graphical elements that allow an easy identification of value information on a coordinate system (see Figure 8.1). Based on the coordinate system, the elements are directly relatable among another. Bar charts are especially useful to compare values against each other. Line charts are useful to identify trends in data. The visualization is well known, thus usually requires little learning. For large data sets, however, bar- and line charts become complex to read.

Bar- and line charts are capable of a simple encoding of time segments and durations in the coordinate system. The displayed shapes can be compared among another. Relations are not visualized, and can only implicitly be deduced based on information objects included in a time segment. Thus, for RO tasks, bar- and line charts are not suitable due to the lack of relation information. For UT tasks, they are only partly suitable, as the relation between the different visualized time segments is not encoded, which complicates the understanding of the process (e.g., once a user wants to know if an information object was used in the beginning of the reviewed time only, he/she needs to check each time segment). For UD tasks, however, the bar- and line charts are suitable, as usage times are encoded in a way that makes them easy to decode and compare.

Examples: Social Wakoopa [129], Rescue Time [131], the CAM dashboard [79] and the student activity monitor [79].

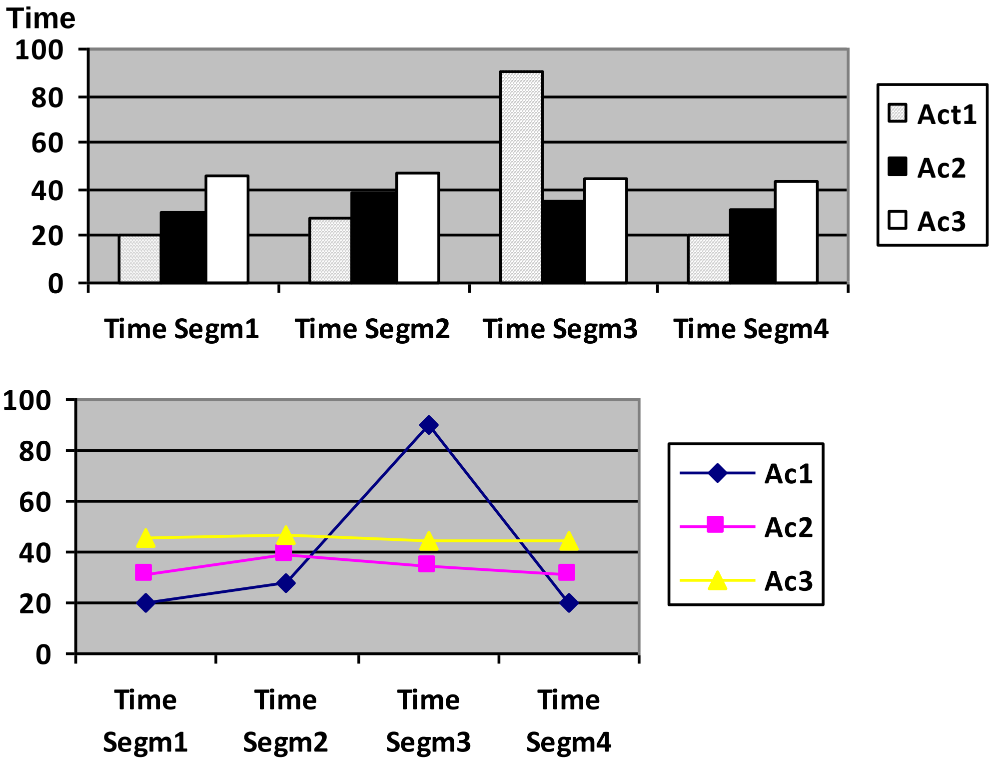

Figure 8.1.: Bar- and line chart visualization of activity data. The amount of time spent (e.g., minutes) with three different activities within four time segments (e.g., days) is visualized.

8.2.3.2 Gantt Charts

When Gantt charts are used for the visualization of activity data, this implies a (two-dimensional) coordinate system, with rows of text identifiers on the y-axis representing one or more activity data elements, and a continuous timeline on the x-axis (see Figure 8.2). Access durations are visualized as blocks with a start, an end and a duration that is expressed by the extent of the block. Relations among elements are visually represented by the proximity or the overlapping of blocks on the timeline. Gantt charts are well-known, therefore, require little learning. For huge data collections, they may, however, become overly complex.

For RO tasks, an encoding of relations is given, although it is not necessarily simple, in particular for elements with long duration. The weight of a relation is the number of similar information object successions, i.e., similar bar successions in the visualization that need to be manually counted. The identification of time segments with Gantt charts is simple, as they are simply encoded in the timeline.

The given aspects make Gantt charts a better choice for RO tasks than bar- or line charts, but they are still complex to read. For UT tasks, a simple encoding of time segments is given. Comparison is complex, as the length of shapes at different y-positions needs to be compared. The process information is well preserved in Gantt charts. One can assume that solving UT tasks with Gantt charts works well, but requires time to compare operations and to search along the timeline. UD tasks are complex with Gantt charts, as the usage duration is spread across the timeline, or encoded in additional text, which makes the tasks solvable, but requires high metal effort.

Example: The Outlook Journal [30] provides a Gantt chart visualization of information object usage. Each information object has a dedicated row. In this row, a bar denotes the usage time of the respective resource.

8.2.4 Interactive Activity Data Visualizations: Novel Types

The reviewed work suggests that the existing visualizations of activity data do not make use of graphs. This is surprising, as there is a need to display elements that have more than one predecessor and more than one successor (e.g., when activity data elements

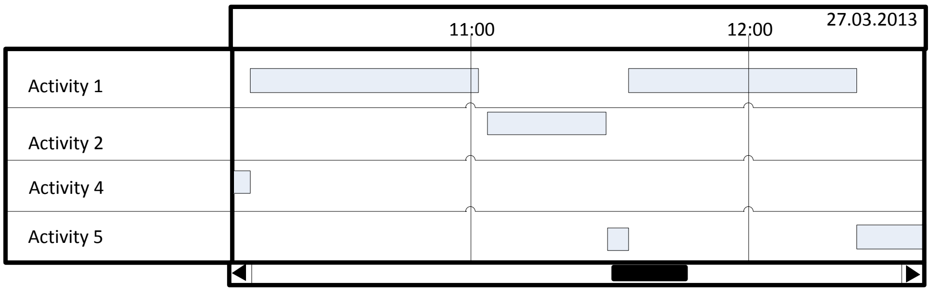

Figure 8.2.: Gantt chart visualization of activity data. The chart shows five different activities and the respective times of work (e.g., the user worked on activity 1 between 11:30 and 12:30 o'clock on the 27th march 2013).

for the same information object are aggregated). To visualize such a structure with numerous connections, graphs are useful, as they apply the law of uniform connectedness[3]. An important requirement for graphs is good readability, even when they contain many elements. Ghoniem et al. [98] studies on nodes with 50 and more vertices show an increasing complexity of graph decoding and understanding.

Two specific types of graphs have gained increasing relevance [151] and are of specific importance for this paper: *dynamic graphs* address the visualization of time in graphs. Element evolution is displayed by the addition or deletion of edges and nodes, e.g., using animation. *Compound graphs* are static graphs organized based on semantic clustering, i.e., a second type of order, e.g., a hierarchy or group is used to organize sets of nodes. Compound graphs are used e.g., in plate notation and UML diagrams.

In the following, the first straight forward visualization of activity data with graphs which was integrated in Transparency 1.0[4] is discussed. The implementation reveals difficulties with respect to readability and temporal understanding of the visualized data, especially when the activity data contains many elements. Based on the gained experience two new visualizations are proposed which address the readability and the visualization issue: the *timeline graph*, a new variant of dynamic graphs, and a *compound graph*, a variant of the hierarchical compound graph applying a hierarchical structure.

8.2.4.1 Limitations of Simple Graphs for Activity Data Visualization

A straight forward graph visualization uses vertices to show activity data elements and edges to show relations between the objects. Temporal information is added as label to nodes. This type of visualization has been integrated into Transparency 1.0 and was evaluated. Nine participants used the research prototype for 2 weeks (for a detailed description of the study setup, see the appendix C.6). This use was accompanied by a series of questionnaires that tracked trends in the perception of the named tool.

Most participants initially expected the graph representation of their work to be useful or very useful (6 of 9). After two weeks in which they used the tool in a normal work context, however, different problems became apparent: the appreciation of the graph representation for the work decreased significantly (not useful (2 of 9), partly useful (2 of 9), moderately useful (2 of 9), useful (3 of 9)). 5 of 9 users considered reading the graph to be very complex. In a subsequent interview, all participants stated that they found the graph view interesting, but did not find a connection to their daily work tasks, and that it was time consuming to interact with the visualization, especially due to its size (after 8 hours of work a graph sometimes contained far more than 100 nodes, see Figure 8.3). Also, the problem of decoding temporal information from the graph was mentioned informally by different participants.

The two visualizations presented below address in particular these concerns.

8.2.4.2 Dynamic Graph With Timeline

The timeline graph addresses the central demands of RO tasks by combining temporal and relation visualization. The lower part of the visualization shows a timeline that displays the periods for which activity data has been logged by a monitoring application (e.g., ContAct monitor). A time segment can be selected in the timeline to investigate in the selected period. For the selected period, the upper part displays a graph of activity data. The graph encodes weighted relations by edges with different thicknesses; vertex size encodes the usage duration of the activity data elements. The period selected on the timeline can be moved to visualize the transformation of the visible graph by animations.

[3] The law of uniform connectedness describes the effect that humans consider elements as related when they are connected by a visual element, e.g., a line. Palmer [206] argues that the law of uniform connectedness is the strongest of all gestalt laws.

[4] The first prototype which resulted from the first user-centered design (UCD) cycle.

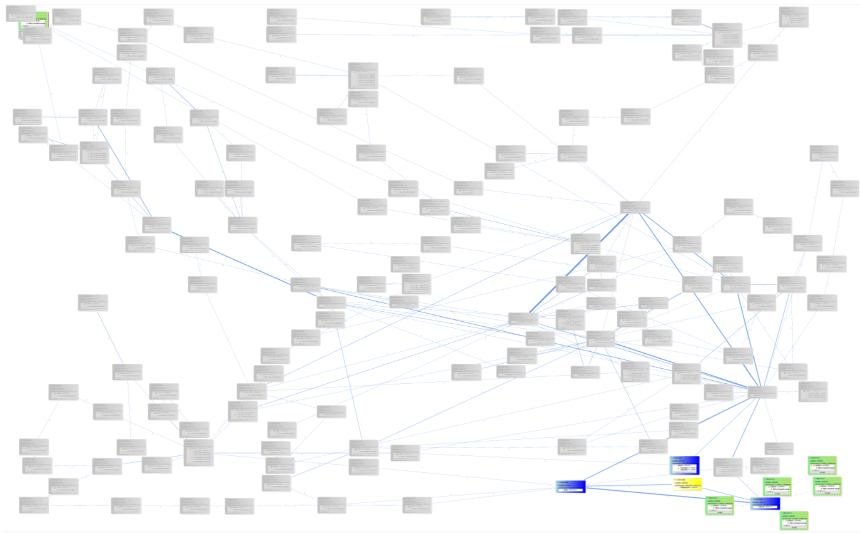

Figure 8.3.: Activity data visualization by a simple graph, after 8 hours of work; layout was calculated by the Inverted Self Organizing Map (ISOM) algorithm. Each node stands for a knowledge action (an application with at least one information object). No knowledge action is repeated. Edges denote switches between knowledge actions.

Time is an explicit organization criterion for the dynamic graph with timeline. The timeline explicitly denotes dates, hours and minutes. This supports a structured access of specific time segments.

The combination of a timeline with a graph addresses the requirements for RO tasks; the encoding of a time segment and comparability based on node size address those of UT and UD tasks. The disadvantage for UT tasks, however, is the way process information is coded in the graph: the user needs to actively change the visible time segment to get an overview of the process. Also, UD tasks may be challenging, as the user needs to identify the time segment that contains the information objects he/she is interested in before they can be compared.

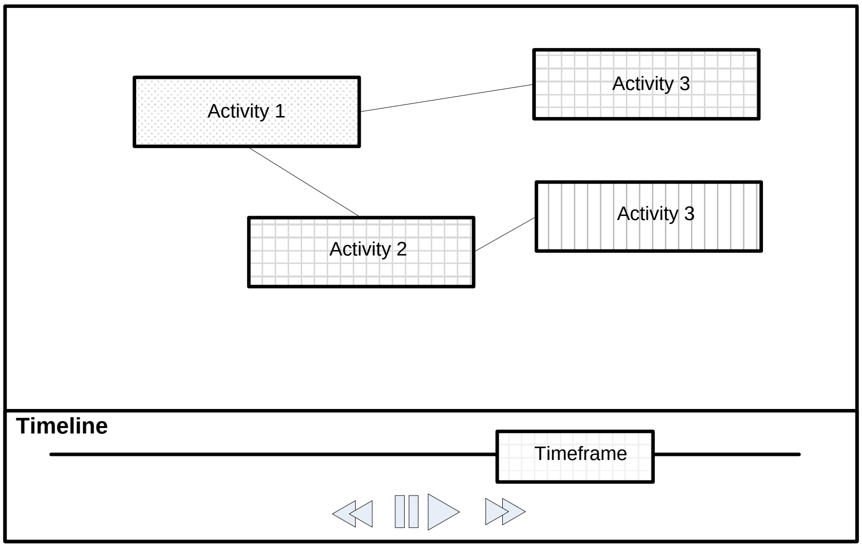

Figure 8.4.: Mockup of timeline graph displaying activity data. Based on the selection of a time segment in the timeline, a graph visualization of the activities performed during the time segment is displayed in the upper part of the visualization. For an implementation of the timeline graph, see the appendix F.6.

8.2.4.3 Hierarchical Compound Graph

The hierarchical compound graph has been designed to address requirements of UD tasks which have not been addressed completely by any of the previously discussed visualizations. In particular, the other visualizations failed to provide an easy means to transfer information about a work process. The hierarchical compound graph organizes the displayed graph with respect to hierarchically ordered groups (see Figure 8.5), i.e., elements are organized in several layers of interconnected boxes that are, in turn, embedded into a temporal coordinate system.

Each box stands for a period of time and contains a graph for interactions. An activity data visualization is only added to a box if all activity data elements were solely used during the period of time that is covered by the box and if it does not fit into the time segment of a smaller box. The y-axis structures the duration time: the longer the box, the longer the duration. The x-axis is a timeline: the x start and end position of the box hint to the length of the displayed period. The boxes are hierarchically ordered. The highest level contains one box which covers the width of the complete visualization. The level below the highest level is decomposed into two equally wide boxes for two shorter periods, standing for the first half of the considered time and the second half of the considered time. The lower level again has twice the boxes, each with a width half the width of the parent boxes, representing again shorter periods.

This organization provides information about a process based on the temporal relatedness of the graphs among each other. Thus a fuzzy understanding of the time segments is conveyed. The highest level introduces an understanding of “always used in the segment”, the layer below introduces an understanding of “only in the first half” and “only in the second half”. A third layer provides an idea of “only in the beginning” and “only at the end”. A decomposition into more than four layers should be avoided as it increases the decoding complexity of the visualization without providing much more useful information (“in the very beginning”, “shortly before the end” are not useful criteria).

The graph fulfills all requirements for UT tasks: time segments are encoded in a coordinate system that transfers process knowledge based on a hierarchical structure. Representation of durations by vertex size enables comparability between elements. The requirements for RO tasks are also met: weighted relations are easy to decode and time segments are clearly displayed. Only for long time segments, the navigation of the hierarchical compound graph is likely to be more complex than for the previously presented timeline graph. Summarized, the requirements for UD tasks are met, although time comparison is presumably simpler using bar- or line charts.

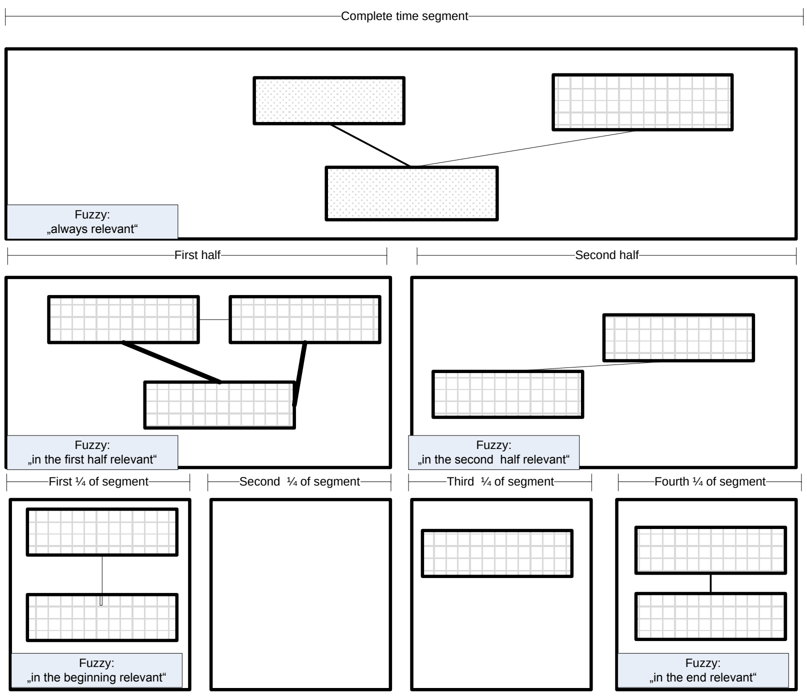

Figure 8.5.: Mockup of the compound graph. The period of time considered is split three times from top to bottom. As a result some boxes show elements which were only relevant in the beginning of the work process during the considered period of time. For an implementation of the compound graph see the appendix F.4.

8.2.5 Interactive Activity Data Visualizations: Comparative Study

While the basic suitability of the respective visualizations for the identified tasks has already been analyzed, a user study is executed to substantiate the claims made.

8.2.5.1 Hypothesis and Study Design

Suitability or usefulness of a visualization for a task is in the following defined by operation success (when a user solves a task with a visualization, the task can be solved correctly, can be solved incorrectly or can be considered as unsolvable with the specific visualization) and time investment. Three hypotheses were posed for the suitability, focusing on the performance of the two proposed visualizations, the timeline graph and the hierarchical compound graph, with respect to the other visualizations:

- ***H1*** The number of errors is lower for a) compound graph and b) timeline graph than for all other visualizations.
- ***H2*** Task completion time is lower for a) compound graphs and b) timeline graphs when compared to bar-, line and Gantt charts for most (at least 4) tasks.
- ***H3*** One of the graph-based visualizations outperforms the other in all tasks (in terms of task completion times and error rates).

To test these hypotheses, six tasks were created; each of them lying in one of the task classes discussed before, and each task class is represented by two different tasks (doc refers to any information object):

- ***Task1*** How much time was worked on *doc*? *(Task class: UD)*
- ***Task2*** When was *doc* used during the work process? (e.g., from 4.00-6.00 AM, from 5.00-5.30 AM, from 4.00-5-00 AM, ...)? *(Task class: UT)*
- ***Task3*** Which documents were connected to *doc* in the work process? *(Task class: RO)*
- ***Task4*** List the documents that were accessed only between 4.00 am and 4.30 am. *(Task class: UT)*
- ***Task5*** You have read an interesting book about patterns when working on *doc*. Can you identify it? *(Task class: RO)*
- ***Task6*** Find 3 resources that were overall used for more than 18 min. *(Task class: UD)*

8.2.5.2 Study Setup

To evaluate the performance of the visualizations on the six defined tasks, a prototypical activity data visualization tool has been created (see Figure 8.6). Initially, the tool shows a questionnaire. Then, the tool asks for the solution of the six presented tasks with the different visualizations. Activity data sets were created inspired by real activity data collected during normal work days.

The amount of resources has been restricted to allow an execution of the study with all 24 tests in 30 minutes. In this configuration, the visualizations display more than 7 discrete elements to assure that activity data size is reflected in the dataset, although the scalability of the visualization with respect to very large datasets is thus not in the focus of the study. Each data set contains 13 resources, and shows 8 to 13 resources at a time. The data shows a time segment of two hours for March 12, 4.00-6,00 AM. Each task has to be executed with every visualization. To rule out learning effects, the tool randomizes among the tasks sequence, the visualization sequence and among four different data sets that are used. The different datasets share the same structure—only task completion times and information object names have been changed. The similarity among the datasets guarantees a similar complexity for the same task solved with different datasets.

To represent a realistic work process, the data sets tackle a focus topic, but also include information objects that belong to different tasks to mimic multi-tasking. The focus topic for the data sets are: 1) Software engineering/UML modeling, 2) Software engineering/Design patterns, 3) Lessings' "Hamburgische Dramaturgie", 4) Eccentric Pump Sales.

For each task, the solution provided by the user and the time spent (in ms) were logged. After the study, each participant was shortly asked to identify the visualization he/she liked the most/least.

Eleven participants were recruited for the study using convenience sampling—10 were male, 1 female, their age was between 25 and 60. All participants use computers frequently during their daily work processes.

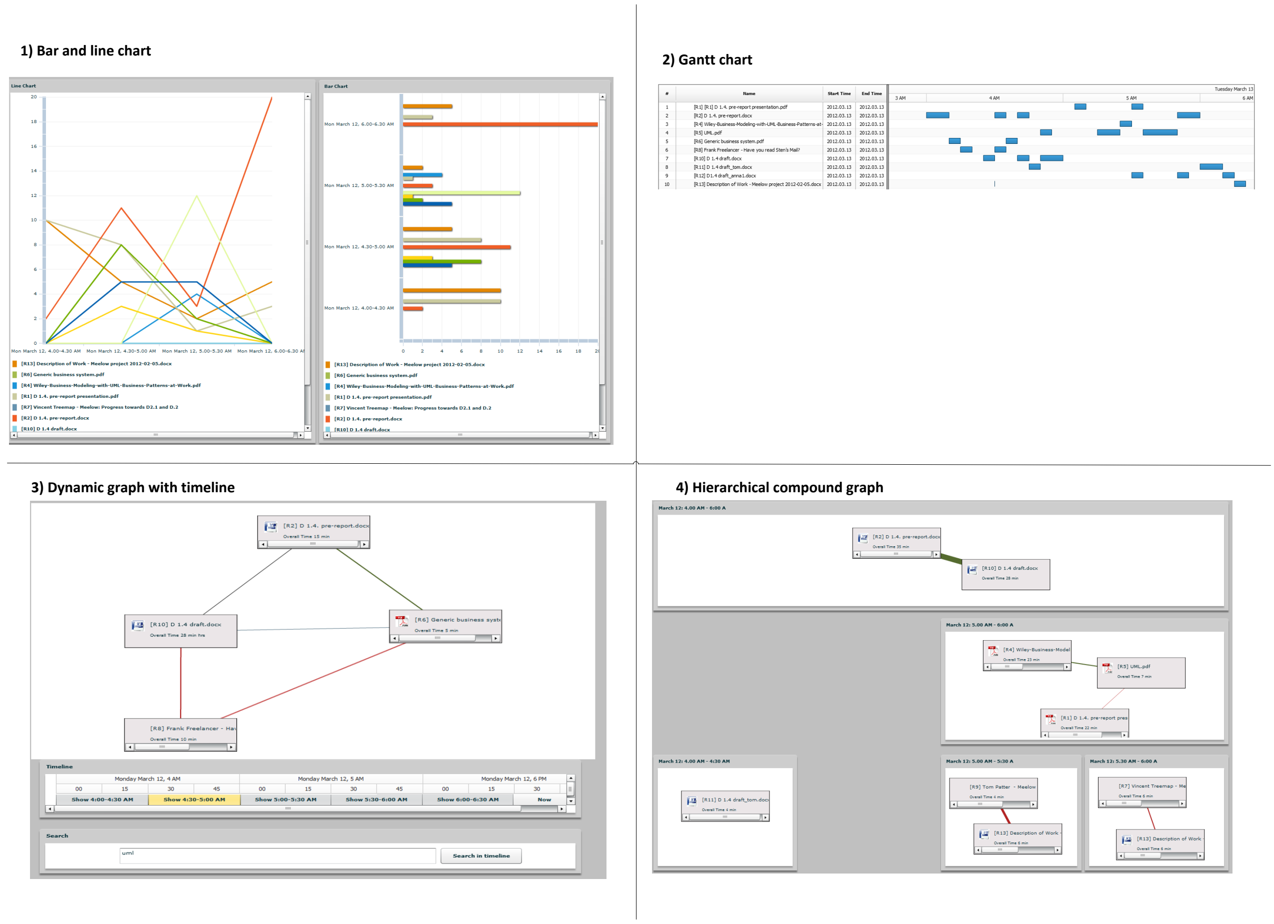

Figure 8.6.: Prototype visualizations of activity data used for the evaluation: 1) Bar- and line chart, 2) Gantt chart, 3) Dynamic graph with timeline, Hierarchical compound graph.

8.2.5.3 Results

The initial questionnaire elaborated on activity data use and process awareness. Ten participants stated that they use history features like timelines or history based auto completion fields during their daily work. Four participants knew the Outlook Journal. With respect to memorization of work processes, 3 stated that they have problems remembering their work (2 not good, 1 okay), whereas 7 stated that they generally can remember their work processes (6 well, 1 good). Nevertheless, no one stated that he could remember all documents he worked on during the morning of the study day (study activities were all performed in the afternoon), 7 stated they remember most, 2 some of these documents—there was, however, no further inquiry to validate these reports. Only two participants stated that they spent little time with searches for documents they accessed earlier, the others spent a considerable amount of time with searches for this type of information (7 some time, 2 much time).

8.2.5.3.1 Number of errors

Each participant completed 24 tasks. Each task could be solved with a correct solution, an incorrect solution, or a note that the task was not solvable with the visualization. The absolute number of errors and rejections for the tasks and visualization is visible in Figure 8.7. It is important to note that a solution existed for each task and each visualization, although the complexity of finding it differed among the visualizations.

- **Line- and bar charts** The charts show the most errors (11) and the most statements that a task is not solvable. Most errors and unsolvable statements occur for task 3 (9 unsolvable, 2 errors) and task 5 (3 unsolvable, 3 errors), which belong to the RO task class. This underlines the problem of visualizing relations in this chart type (they are only implicitly encoded in the time segments). The UT and UD tasks show fewer errors, without being considerably good results. The difficulties for UT and UD relate to the complexity of bar chart reading for many elements.

- **Gantt charts** The Gantt chart showed 13 unsolvable statements and 11 errors. The participants had problems with RO tasks in particular. Although relations are encoded in Gantt charts, the identification of the relations among the different rows is error-ridden, and sometimes even discarded by users due to its complexity. Considering the three task classes, the Gantt chart performed best for UD tasks, as process information is visible based on the timeline. UT tasks showed difficulties, as the users had to identify all bars for each row to identify usage times.

- **Timeline graph** The timeline graph showed good results for all tasks with no unsolvable consideration and only five errors. The errors mainly occurred for task 4, a UD task. As the timeline graph does not include a simple encoding of the process, the participant needs to identify the work process on the period successions in the timeline which is complex and error-ridden.

- **Compound graph** Overall, the hierarchical compound graph showed the best results. Only one error and one unsolvable statement occurred for task 4.

		Task1	Task2	Task3	Task4	Task5	Task6
Barchart	unsolv	1	2	9	4	3	3
	Error	4	1	2	0	3	1
	TRUE	6	8	0	7	5	7
Gantt Chart	unsolv	0	0	5	3	3	2
	Error	1	0	5	2	3	0
	TRUE	10	11	1	5	5	9
Compound Graph	unsolv	0	0	0	1	0	0
	Error	0	0	0	1	0	0
	TRUE	11	11	11	9	11	11
Timeline Graph	Error	0	0	1	4	0	0
	TRUE	11	11	10	7	11	11

Figure 8.7.: Correct solutions, false solutions and “unsolvable” notes (per task).

Summing up, H1a and H1b can be confirmed, as the timeline graph as well as the hierarchical compound graph overall showed better results with regard to the number of errors than the other visualizations.

8.2.5.3.2 Usage time

All tasks were executed between 7000 ms and 120000 ms (see the scatter chart in Figure 8.8). To make statements about the time distribution among the different visualizations, significance needs to be tested, e.g., using an ANOVA test. This requires a normal distribution and variance homogeneity.

To test for normal distribution, the Shapiro-Wilk test is applicable for a data set of the given size. During the study execution some people started to execute tasks before they understood them and spent time to think about the task. This produced outliers which were eliminated following the three sigma rule (99.73 % of the values lie within 3 standard deviations of the mean) and replaced by the mean value (cf. [160]). Shapiro-Wilk shows that a normal distribution for an alpha level of 0.05 can be assumed for all but two datasets (the data for the dynamic graph in task 3 and the data for the bar chart in task 4). Subsequently the Levene test for homogeneity was applied, finding that the homogeneity is acceptable. For task 5 homogeneity assumption needs to be rejected.

As most data is normally distributed and variance homogeneity holds for all but one distributions the application of an ANOVA test is valid (cf. [148]). Only task 5 was excluded, as homogeneity was rejected.

- **Compound graph vs. classical charts**

  The hierarchical compound graph, the bar/line chart and the Gantt chart time series per task (rows) and visualizations (columns) have been used as the input for a two factor ANOVA with replication. The result shows significance columns. The value for the columns is of interest as this describes the difference between the different graph types ($F(3,06) = 96{,}41$, $p < 0.001$).

  Based on the average task completion times, one can investigate this further. In table 8.2 the average values of the task completion time per visualization and task are given. Based on this information, the strength and weaknesses of the different visualizations can be identified. It is striking that the hierarchical compound graph outperforms the other visualizations (=lowest average value) for all tasks, except task 4. The identification of usage time and usage duration seems to be simple with the graph. The hierarchical compound graph seems to fit the requirements for UT, UD and RO tasks very well. Only in some cases, like the UD time segment identification of task 4, which is straight forward for the time segment in question in that task, bar/line charts show their strength.

  Summing up, H2a can be confirmed.

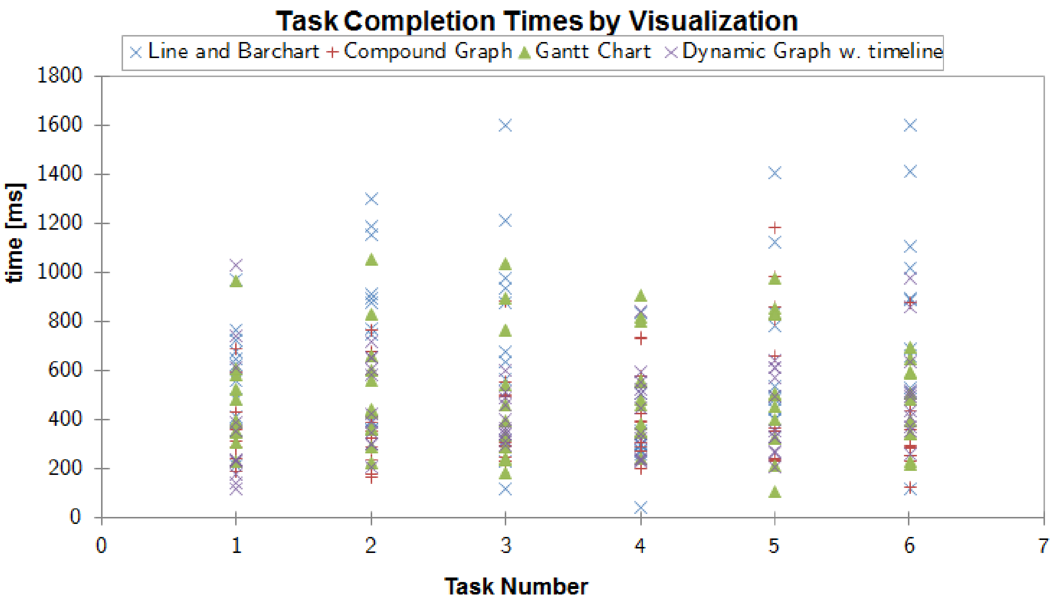

Figure 8.8.: Task completion time (per task).

Task	Line and Bar-chart	Gantt Chart	Compound Graph
1	56446.27	39735.53	26446.27
2	87247.92	47628.09	30801.65
3	75305.77	37942.14	31446.28
4	30008.27	43099.16	32181.81
6	77801.64	45619.82	28454.54

Table 8.2.: Distribution of the average values for compound graph vs. classical charts in ms.

- **Timeline graph vs. classical charts**

  The timeline graph, the bar- and line chart and the Gantt chart time series per task (rows) and visualizations (columns) have been used as input to a two factor ANOVA with replication. The result shows significance for columns which means that there is significant difference between the used visualization ($F(3,94) = 13,41$, $p < 0.001$).

  Again, the average values of the time spent with each visualization for each task are considered (see Table 8.3): the average values are better for all tasks, except task 4 and 6. These tasks ask to identify usage time and usage duration. The timeline graph performs especially well for RO tasks. Although the results for UD and UT tasks are less positive, they are still convincing.

  Summing up, H2b can be confirmed.

- **Timeline graph vs. Compound graph**

  To compare both graph visualization, the respective time series per task have been used as input for a two factor ANOVA with replication. The result shows significance for the columns. The effect of visualization types (columns) on task completion time gives: $F(3,94) = 13,41$, $p = 0.0004$. The null hypothesis that the values are significantly different can be accepted. The average values of the hierarchical compound graph are better than the results of the dynamic graph with timeline for all UT and UD tasks.

  The graph-based visualizations show very good results with respect to task completion time and error rate for all considered task classes. Still, the study does not allow a decision on one visualization which performs better for all tasks (no significant difference between the two graph-based visualizations with regard to task completion time). Therefore, H3 needs to be rejected.

8.2.5.3.3 Post-test interview

After their trials, participants were asked for the visualization they appreciated the most/the least. With the exception of 2, everyone considered the compound or the timeline graph as the most suitable visualization; bar- and line charts were the least appreciated

Task	Line and Bar-chart	Gantt Chart	Dynamic Graph
1	56446.27	39735.53	25958.67
2	87247.92	47628.09	44041.32
3	75305.77	37942.14	36082.64
4	30008.27	43099.16	36256.2
6	77801.64	45619.82	45504.12

Table 8.3.: Distribution of the average values for timeline graph vs. classical charts in ms.

Method Design	
Basic principle	• Mediate the creation of memory cues based on activity data
Design directions	• Exploration • Organization • Recommendation
Extraction tasks	• UD task (identify Usage Duration) • UT task (identify Usage Time) • DC task (retrieve by DesCription) • RO task (find by RelatiOn)
Interaction design	
Basic guidelines	• simple, easy to learn • permanent access and high responsibility • efficient • user acceptance
Interactive visualizations	• List • Set • Bar-, line-, Gantt chart • Dynamic graph with timeline • Hierarchical compound graph

Table 8.4.: Design space components.

visualization. This is very much in line with the results given above: the graph visualizations are less prone to error, and have significantly smaller task completion times.

8.2.6 Intermediate Results

The interaction design guidelines structure the decisions involved in the development of a visualization method to deal with activity data. The constraints of interaction (e.g., responsiveness, user acceptance) and the selection of a visualization based on the information extraction tasks are specified. Intensive research on the mental effort of information extraction from visualizations has been conducted. To address a lack of visualization techniques which consider associative extraction tasks, two novel interactive visualizations were developed, namely the dynamic graph with timeline and the hierarchical compound graph. An evaluation of those two new methods has shown that they show good for most information extraction tasks. The following hypothesis were confirmed for the specified tasks:

- The number of errors is lower for a) compound graph and b) timeline graph than for all other visualizations.
- Task completion time is lower for a) compound graphs and b) timeline graphs when compared to bar-, line and Gantt charts for most (at least 4) tasks.

The developed visualizations will be used within the support methods developed in the next chapter of this thesis. The final evaluation will show that the methods which use these visualizations decrease the likelihood of memory failures in information work.

8.3 Summary

This chapter has specified a design space for information work support methods. The design space is a necessary preparation for the design of user support methods because a direct realization of the identified requirements is infeasible. Thus, the design space bridges a gap between requirement identification and method design and facilitates method development by providing necessities and decisions involved in the design process. In this respect method design includes the decision for a design direction and the choice of the extraction tasks to be supported. For the direction and the selected extraction task an appropriate (low decoding effort) visualization is selected. The design space is composed of guidelines for method and interaction design. For an overview of the respective components, see Table 8.4.

The next chapter showcases three support methods which follow the guidelines of the design space. The design space will additionally facilitate the classification and the comparison of the proposed methods.

9 Information Work Support Methods: Showcase

The overarching goal of this thesis is the support of information work to limit the likelihood of memory failures. This chapter introduces three methods to address memory failures in information work based on activity data. An evaluation shows that the methods decrease the likelihood of memory failures. Thus, the evaluation results indicate that activity-centric user support is a successful answer to the research question “How to limit prospective and retrospective memory failures in information work at the computer workplace?”.

The following methods have been designed:

- **Activity-centric task management (organization):**
  - *Description: Activity-centric task management* provides an overview of a subject’s tasks (see section 9.1). Activity data facilitates the creation and maintenance of the task objects and is used to provide work process information.
  - *Addresses:* The method addresses memory failures related to the work organization of unfinished activities and provides an overall activity awareness. Therefore, the method supports the organization of unfinished activities to provide an overview of the existing work processes and to facilitate activity switches.
- **Interactive activity history (exploration):**
  - *Description:* The *interactive activity history* gives access to activity data as network of knowledge actions organized by a timeline (see section 9.2). The subject’s history is explorable based on a browser with filter and search capabilities.
  - *Addresses:* The exploration of the activity history addresses forgotten activities and related elements. Therefore, the method facilitates the creation of memory cues to recall the work process and involved information objects (cf. section 9.2).
- **PASTREM activity centric recommender (recommendation):**
  - *Description:* The *PASTREM activity centric recommender* generates proactive recommendations of information objects based on the most recent work process of a subject (see section 9.3).
  - *Addresses:* The method addresses two problems. On the one hand the identification and access of information relevant for a work process which is resumed after an interruption. On the other hand support for the switching between different activities. To achieve these types of support, the system can be configured to recommend information closely related to a subject’s work process for focused work execution or to recommend information of general relevance for multitasking oriented work execution.

The three methods have been developed by executing two user-centered design (UCD) cycles (cf. section 5). All methods use interaction data processed and analyzed based on the processes described in the chapters 6 and 7. Each method realizes one of the three design directions of the design space to mediate a subject’s recall process (cf. chapter 8). As a proof of concept, the methods have been realized and integrated in one application which is referenced as Transparency 2.0 in the following.

Next to the work on memory failure likelihood this chapter addresses another goal: The chapter showcases the usefulness of activity theory based system design method (AT-SDM) in UCD. The designed methods result from two iterations of the UCD cycle initialized in chapter 5. In favor of a clear and straight forward result presentation the cycle characteristic of the development is not explicitly highlighted in the presentation of the methods. Information about the UCD process is provided in the second to last section of this chapter (see section 9.4). First, the methods developed during the first UCD cycle and their evaluation are presented. Second, the information gain from the first UCD cycle is considered to adapt the context of use.

The chapter concludes with an evaluation of the support methods’ effect on memory failures (see section 9.5).

9.1 Organization: Activity-centric Task Management

Activity-centric task management addresses memory failures related to the work organization of unfinished activities and provides an overall activity awareness.

Task management is a very useful technique of organizing work processes. The state of the art review has shown that the basic disadvantage is the high manual effort required to create and maintain the tasks according to the real work process. Activity-centric task management reduces this effort by using activity data for task creation and maintenance. Questions like “What do I still have to do?", "How much time did I work on writing the sales report?" or "What do i need to continue working on the sales report?" are answered. Work related to this method has been discussed in the state of the art review (see section 5.3).

This section gives a detailed overview of activity-centric task management. First, the taken method and interaction design directions are presented based on the design space (see section 9.1.1). Second, the task management process and the respective user interface is provided (see section 9.1.2)[1].

Finally, the integration of the activity-centric task management is illustrated by two example scenarios (see section 9.1.3).

9.1.1 Design Space: Method and Interaction Design

In the following, the taken directions for method and interaction design are reported. This gives a first overview of the method in terms of the design space and helps to classify the method.

9.1.1.1 Method Design

The idea is that tasks are created and maintained based on activities mined from the subject's interaction history. Because of the information included in the mined activities (e.g., duration of activity, work process, information objects) task management is enriched with additional information.

The use of activity data addresses the pain points of existing task management systems, namely the complexity of maintaining the structure and user's reluctance to manage large amounts of data manually (e.g., task names, due dates, etc., cf. section 5.3).

Task management is an active creation of memory cues. By creating task objects, the subject manually creates a structure which is intended to support the work organization by externalizing tasks, constraints of tasks and involved objects. The structure helps to remember what needs to be done and facilitates task switches because the involved applications and information objects are accessible. In particular, the requirements of "should help derive existing activities" (RQ1, Tension T1), "should help derive activity related elements" (RQ2, Tensions T2, T3, T4) are addressed. Additionally an overview of the "executed work process" (RQ4, Tension T2, T3, T4) is provided. A maintained task management system "should support activity switches" (RQ5, Tension T6).

The following design direction with respective extraction tasks is valid for activity-centric task management:

- **Design direction:** The basic idea of a task management extension suggests the organization design direction. Managing tasks means organizing information. Activity-centric task management similarly focuses on the organization of data. Thus, functions of accessing activity data, enhancing it with metadata and accessing the data are required.

- **Extraction tasks:** Task information is an externalization of unfinished work. UD task (identify Usage Duration) and UT task (identify Usage Time) are supported by the usage duration information included in activity data. The textual content included in activity data supports the DC task (retrieve by DesCription).

9.1.1.2 Interaction Design

In the following, the chosen interaction design for activity-centric task management is described:

- **Basic guidelines:**
  - *Simple, Easy to learn:* The creation and maintenance effort of tasks is reduced to a selection process. The user accesses a list of mined activities and selects those activities relevant to be followed up.
  - *Permanent access:* The method needs to be implemented in a software which is running permanently in the background and which can be accessed without much effort (e.g., by using the Windows Jumplist or a tray icon).
  - *User acceptance:* Task management is a well-known work organization technique. Due to its widespread usage in different applications most users are familiar with the technique. Activity-centric task management does not change the way task management is conducted but is designed to unobtrusively extend well-known functionalities. Familiar elements help to increase the perceived usefulness to create an intention of use [283]. Once technology acceptance is given, users are willing to explore the additional features and the tool may replace existing work organization techniques.
  - *Efficiency:* To enable rapid and successful interaction a clean and rigidly structured user interface is focused, following two interface paradigms: in-place editing and details by expansion.

[1] In this section, only concepts are shown. For images of the real implementation of the activity-centric task management, see section F in the appendix.

* IN-PLACE EDITING: In-place editing refers to the modification of elements without using dialogues or menus. By overlaying selections and input fields and integrating drag and drop functionalities, the user can make all modifications in the main visualization.
* EXPANSION: Expansion refers to the access of detailed information by expanding elements of interests. Expansion allows the quick access to detail information while the element of interest remains embedded in the group it belongs to. In Transparency 2.0 each task element in the task list can be expanded to access related activity data.

- *Visualization choice:* The main element of activity-centric task management is a task list. To provide information about the work process of an activity, the hierarchical compound graph is used additionally (see section 8.2.4.3).

9.1.2 Process: Task and Activity Management

Activity-centric task management carefully extends the established task management paradigms. The commonplace techniques of creating task objects with metadata and their organization in a task list remain unchanged. Activity-centric task management adds additional information to each task and adds a user interface to create tasks based on mined activities. The respective processes of task organization and task creation based on activity data are described in the following.

9.1.2.1 Task Organization in an Expandable Task List

The creation and maintenance of tasks is realized based on an expandable task list. The list gives an overview of existing tasks and provides access to the activity data of each task (see Figure 9.1). Each task object contains a list of information objects, statistics of usage time and a process visualization based on knowledge actions:

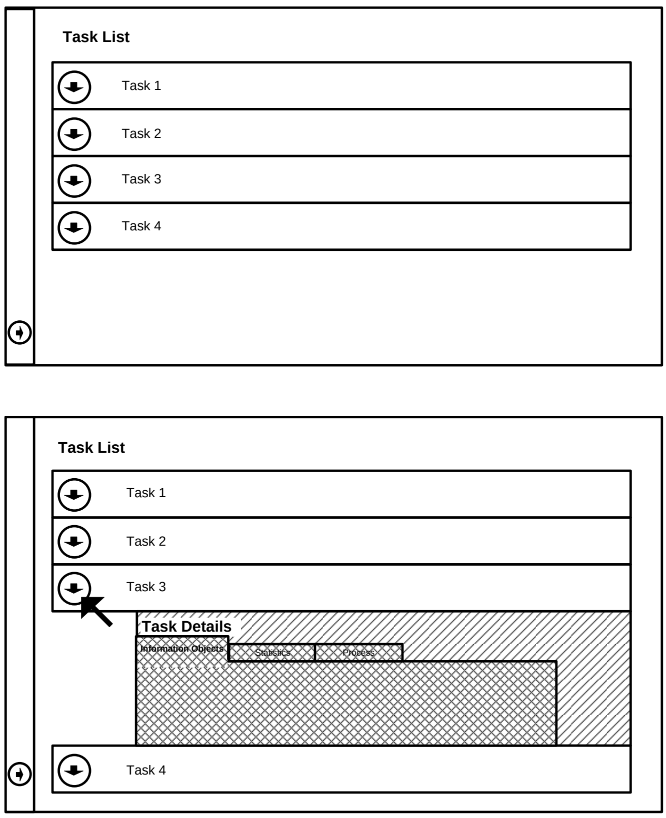

Figure 9.1.: Task list and item expansion to access task details. For an implementation, see Figure F.1 and F.2.

- **Information Objects:** Each task contains a set of information objects that are relevant for the task execution (see Figure 9.2, a).

  The set provides quick access to relevant information objects. The font size of a resource shows the time spent with it compared to all other resources (thus indicating relevancy). Each element can be opened by a double click. A play and a stop button help to activate or deactivate tasks by opening or closing all elements that belong to the task. Each information object is connected to an activity data set which holds information about the knowledge actions and the desktop operations the object is involved in. If activity data for an information object exists, not only its interaction type but also the time it was already used is provided to the user. New information objects can be added to a task, using drag and drop. With the integration, the information object is tracked for the task and the usage time is maintained accordingly.

For the visualization information objects which belong to the same base URI are grouped, i.e., websites from similar URLS and files from similar folders are grouped together. This addresses problems of users to read tasks which include very many information objects. The duration is used to order the cluster based on relevance.

- **Statistics:** A bar chart is used to show the distribution of the work on the task over different days (see Figure 9.2, b). For each day the distribution of time among the different information objects included in a task is visualized. Additionally, the fragment of different knowledge action types is provided (e.g., 30 percent authoring, 50 percent consuming, 20 percent communicating).
- **Work process visualization:** The task visualization shows the execution process using the hierarchical compound graph (cf. section 8.2.4.3)). The graph helps the subject to build a fuzzy process knowledge (see Figure 9.2, c). Answers like "What was done in the beginning?" or "Which information objects were used during the whole work process?" can be answered directly based on the visualization.

The task list only gives an overview based on task names. The identification of a task based on remembered information objects is complex without expanding each task element. This problem is addressed by a search function among the tasks and their included knowledge actions with attached information objects.

The task data is frequently updated based on the monitored activities. This assures that detail information in the task list, the duration data, the process and the statistics are maintained.

9.1.2.2 Activity Data

The tasks which are maintained in the task list can be created in a semi-automatic process based on activity data. The user can access a list of mined activities. The list is populated with activities created by an activity mining processes as described in chapter 7.

The user reviews the mined activities and is able to create task objects based on activities. In this sense, the creation of new tasks becomes a review process. It is only in the case that an activity was not yet started that it is not considered automatically. For all other activities an automatic representation is generated based on activity mining. Each mined activity is a cluster of knowledge actions, including respective information objects, applications and detailed work process data. Thus, the detail information accessible in the task list is generated automatically.

9.1.2.3 Mined Activities User Interface

Building tasks from mined activities as a review and selection process has been integrated in the task list user interface. The activity mining process runs in fixed intervals and generates new mined activities and updates the saved tasks (new situations are added to the managed knowledge actions to update the respective time). Each time the mining process has finished, a notification is given to the user that the mined activities have been updated. The user can access the mined activities by expanding a respective pane in the task list view (see Figure 9.3).

As users work on very many different activities during the course of the day and the resulting mined activities set can become very large. To address this, the mined task list can be grouped and filtered based on age and duration. The age refers to the age of the latest included interaction and is grouped in fuzzy day categories: yesterday, last week, last month, older. The duration refers to the overall time the user spent with an activity. The duration is also grouped in fuzzy time categories like up-to 10 minutes, 10-30 minutes, 30-60 minutes, 1-5 hours, longer. Another support of the review process is the cluster name. Techniques of cluster labeling based on mutual information and information gain can work on the knowledge action content of the mined activities to create task names [46]. Finally, a search process allows the identification of activity clusters that contain specific information objects.

Each mined activity can be expanded to access additional information about the included information objects, the work process and the distribution of the time spent with the task.

Creating a task from a mined activity is realized by a selection which enables in place editing options to remove or add elements and to change the name if needed. Based on drag and drop a manual transfer between existing tasks and mined activities is possible as well.

9.1.3 Summarizing Scenarios

The tensions identified are addressed within different interaction scenarios of the task list:

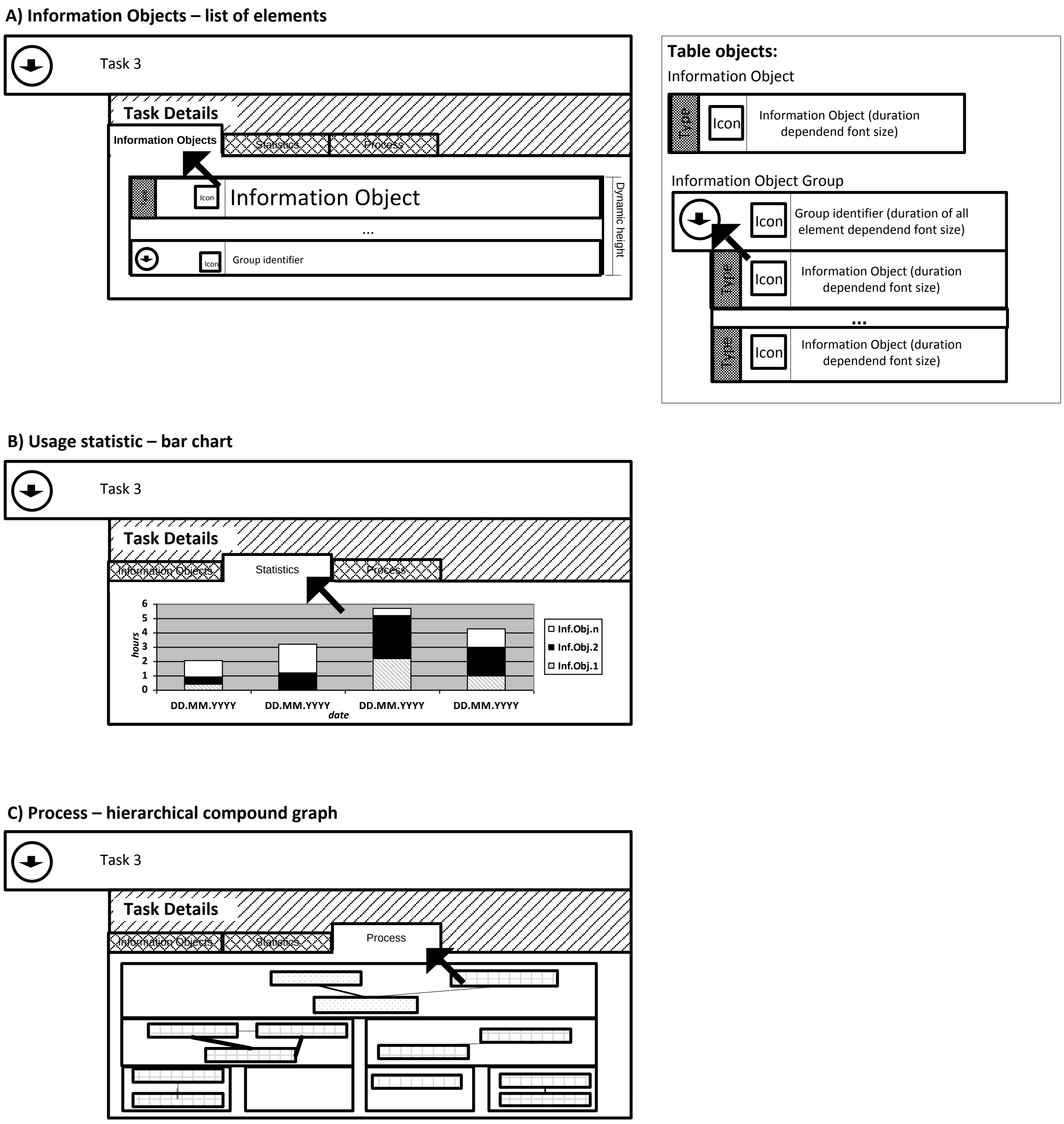

Figure 9.2.: Expansion of a task element gives access to three visualizations based on activity data: a) information objects with use times b) usage time statistics per information object c) process visualization based on the hierarchical compound graph (cf. section 8.2.4.3). For an implementation see Figure F.2, F.3 and F.4.

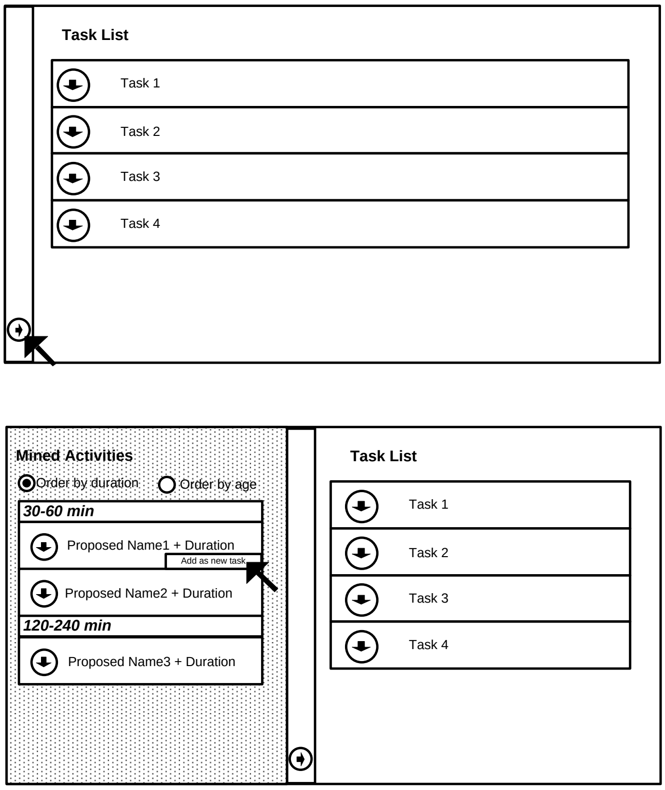

Figure 9.3.: Accessing mined activities to create new task elements. The implementation is visible in Figure F.5.

- **Scenario 1 – Get overview of activities:** To get an overview of all existing activities, the information worker first checks his task list. For each task information about the last interaction and the respective work process is accessible. To check for further activities, the mined task tab can be opened. By searching or grouping mined activities an overview is provided. Those activities which have a high relevance can be foundation for a new task object in the system.

- **Scenario 2 – Switch between activities:** To switch between activities which are captured in the task list, few interactions are required. In the simplest case, the information worker makes use of the start/stop button. For the running task, the stop button is hit. All related information objects which are opened are automatically closed as a consequence. For the upcoming task the start button is hit and the respective information objects are automatically opened. If only some information objects of the new activity are known, the search feature can be used to check if a respective task or mined activity exists which can be used to start the work.

The scenarios show that activity-centric task management adds a historic perspective to tasks. Tasks are generally future directed as they are externalized anticipations of objectives, thus helping individuals to keep track of activities and to plan work processes (cf. section 3.1). Activity data is past directed, it records what was done and adds information about usage duration, usage time and the work process structure to the task. In this respect activity-centric task management connects historic work processes, the anticipation of upcoming work processes and facilitates the creation of a work process awareness.

9.2 Exploration: Interactive Activity History

The interactive activity history addresses memory failures based on the exploration of a work history. The history is intended to provide a temporal and a structural understanding of the work process: what happened when and what was related with what. Emerging memory failures can be directly addressed by exploring the history based on the remembered facts. In this respect, the history helps to get answers to questions like “Which website did I read when I worked on the sales report?” or “What did I do yesterday morning?” The interactive activity history is especially related to the work on activity specific information access and activity awareness reviewed in section 5.3.

The major challenge of an exploration of the work process is to provide access to activity data. Activity data of a longer time span is a complex and very large data structure. The data needs to be made accessible in a way that a subject is enabled to recall facts based on the data. For the interactive activity history this challenge is addressed by a hybrid visualization which combines a graph and a timeline to visualize activity data (the dynamic graph with timeline, presented in section 8.2.4.2).

The interactive activity history is presented in the following. First, the method and interaction design decisions are presented which gives a basic understanding of the way the identified requirements are addressed (see section 9.2.1). The use of the method within an exploration process is described next (see section 9.2.2). The section concludes with a summarizing scenario (see section 9.2.3).[2]

9.2.1 Design Space: Method and Interaction Design

In the following the taken directions for method design and interaction design of the interactive activity history are reported.

9.2.1.1 Method Design

The interactive activity history gives access to activity data as network of knowledge actions (see chapter 6). For this data structure a time and object based exploration of a subject's activity history is realized. Time based exploration refers to accessing a work process visualization for a specific time segment. Object based exploration refers to accessing a fragment of the work process which is connected to the use of a specified information object.

The method addresses an underspecified information need. Based on few remembered facts (e.g., time and object) a subject accesses activity information. The visualization of relations is intended to help derive activity elements (RQ2), connections of activities (RQ3) and the executed work process (RQ4) as well. To some extent, the exploration also supports activity switches (RQ5).

The following design direction with respective extraction tasks is valid for the interactive activity history:

- **Design direction:** The design direction is exploration. Data exploration demands functionalities to show relations (e.g., the relations between knowledge actions). Functionalities to identify and follow relations to explore new parts of the data set are required. A closer investigation of the exploration is provided in section 9.2.2.

- **Extraction tasks:** The proposed approach especially addresses the problem of activity relatedness (by providing a history of connected knowledge actions) and the complexity of information encoding (by using a visualization technique evaluated with respect to its suitability for the intended tasks). In this respect, the find by relation (RO) and identify usage time (UT) tasks are in the focus of the approach.

9.2.1.2 Interaction Design

In the following, the chosen interaction design for the interactive activity history is described:

- **Basic guidelines:**
  - *Simple, Easy to learn:* The use of the interactive activity history builds on well-known visualization and interaction paradigms. Graphs and a timeline are used and can be manipulated directly.
  - *Permanent access:* The method needs to be implemented in a software which is running permanently in the background and which can be accessed without much effort (e.g., by using the Windows Jumplist or a tray icon).
  - *Efficient:* The data access needs to be on a level of abstraction useful to support a subject's recall processes. Tests with different configurations have shown that a low degree of detail (event or desktop operation level) is not useful to support the exploration of the work process. The level of knowledge actions turned out to be a useful level of granularity.
  - *User acceptance:* The method needs to be clearly structured and should not show too much information at once.

- **Choice of visualization:** To realize the interactive activity history the dynamic graph visualization with timeline was chosen. The dynamic graph with timeline or the hierarchical compound graph are possible choices which address both the temporal and the structural aspect. Both performed well for the considered extraction tasks (cf. section 8.2.5) while the hierarchical compound graph visualization even performed slightly better than the dynamic graph with timeline.

  Two reasons lead to choosing the dynamic graph with timeline for a history visualization which organizes large amounts of activity data:

[2] In this section, only concepts are shown. For images of the real implementation of the interaction activity history, see section F in the appendix.

– *Reason 1–Process type:* First, the hierarchical compound graph follows a logic of hierarchical decomposition based on the use of a knowledge action in a time segment: the highest level contains only those elements which were used during the complete time segment. The level below decomposes the elements into those elements which were only used in the first half and those only used in the second half of the considered time segment, etc. The result is a fuzzy understanding of the time by criteria like "done before", "done after", "always relevant in the considered time". The timeline of the dynamic graph provides an explicit presentation of time by dates and hours, supporting the selection of time segments of interest. The disadvantage of the dynamic graph is a loss of understanding for the temporal structure within the selected time segment.

  The explicit temporal structure of the timeline was considered beneficial for the visualization history data. This is one reason for choosing the dynamic graph with timeline.

– *Reason 2–Amount of data:* The hierarchical compound graph visualizes all data directly in the hierarchical structure. For large amounts of data a hierarchy of many levels is required to get a useful granularity and to avoid placing very large graphs within each level. Navigating such a large structure, even when considering features like search and filter capabilities is difficult. This is the second reason for choosing the dynamic graph with timeline.

9.2.2 Process: Data Exploration

In the following, the overall data exploration process realized in the interactive activity history is described. Based on the components of the history data exploration methods are described. The interactive activity history is composed of three elements (see Figure 9.4):

- **Timeline:** The timeline allows the user to select one continuous time segment for which a graph will be visualized. The time segment can be moved by fixed time intervals and the graph changes accordingly by animated adding and removing of nodes. The timeline integrates different time scales: hours, days and weeks. Two overlays over the timeline exist. One overlay which shows the time segment the user is actually looking at. The other denotes the time for which activity data was collected.

  If the user does not select a specific time segment, the timeline visualizes the most recent activities. This allows the user to quickly identify the last work situation. If the system was shut down, the graph helps to recall the last activities and to (re-)access all relevant elements.

  Additionally, hints to the details of time segments can be added to the timeline, e.g., the dominant knowledge action type or information about the amount of accessed information objects within time segments.

- **Graph visualization:** The graph is composed of connected nodes of knowledge actions connected by edges which denote the switch frequency between the elements. Only those knowledge actions are shown which had activities in the time segment selected in the timeline. Each knowledge action node contains one or more information object. To provide a homogeneous presentation and simplified overview, all nodes are initially shown similarly in a collapsed mode (see Figure 9.5). In collapsed mode, the visualized content is limited due to the fixed width and height of the nodes. The provided information in collapsed mode addresses the node type, an information object with an icon, information about the time the knowledge action was active and the number of hidden elements. Once the user hovers over a node, the node is expanded and shows all information objects included in the node. By double clicking on information objects, these are directly opened.

  Knowledge action duration and switch count are provided in expanded mode for the selected segment as well as for the complete available data. The selected segment information provides data for the selected period: "How much time was spent in the selected time segment with the knowledge action represented by the graph?" and "How often were switches between two knowledge actions performed within the specific time segment?" The overview visualization considers all data collected by the tool, including the time spent in total with a knowledge action and the overall amount of performed switches between the visualized knowledge actions.

  Node colors help to relate the visualized graph to the latest user activities. If the knowledge action the user last worked on, is part of the graph visualization, the node is highlighted by color.

- **Filter and search:** Based on filtering nodes are removed from the graph which do not comply with selected filter criteria. This use of filters helps to improve the overview and get a better understanding of the visualized work process graph. Filter criteria which address the age of knowledge actions and their duration have proven useful in this respect. Limiting the visible knowledge actions to those which have not been inactive for less than a given amount of time only shows those elements which remained relevant for the work process at a later point in time. Limiting the visible knowledge actions to those which have been active for more than a selected time segment limits the graph to those knowledge actions which have a high overall relevance. The filters—used alone or in combination—help to limit the complexity of the graph visualized and to focus the visualization on relevant elements.

A search allows the quick identification of time segments an element was active in. Search results are visualized in the timeline and in the graph presentation. In the timeline, an overlay shows regions the searched element has been active in. In the graph, those nodes which are related to the search are highlighted.

A neighbor based exploration allows a quick limitation of the visible elements. By selecting one node, only those nodes remain visible which are direct neighbors of the selection. This is especially useful in combination with the search, as based on an identified node of the search, the graph can be focused and then explored on demand.

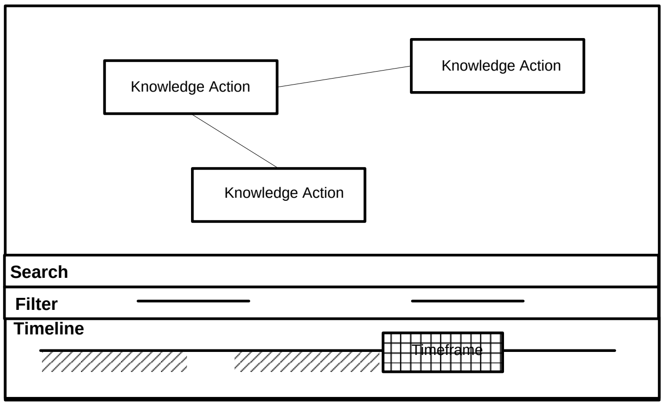

Figure 9.4.: Overview of interactive activity history components: 1) Graph 2) Search and Filters 3) Timeline. For an implementation, see Figure F.6.

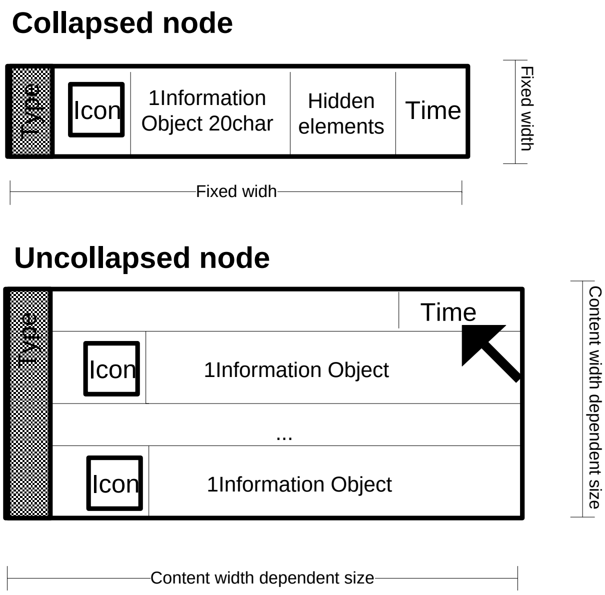

Figure 9.5.: Node design collapsed and uncollapsed.

9.2.3 Summarizing Scenario

In the following, two scenarios are presented which showcase the use of the interactive activity history to access memory cues 1) based on time search and 2) based on object search. Although the time and the object based scenario are presented separately, a combination of object and temporal exploration is supported by the system as well.

9.2.3.1 Object Based Exploration

Object based exploration relates to the use of the activity history to identify situations an object has been relevant in. Work process recall and associative search are two scenarios object based exploration can be used for.

- **Object based work process recall:** In a bootstrapping-like process object memory can serve as an entry point to recall work process knowledge. Such a recall process is supported by object based exploration. The subject enters a remembered object's name in the search field and those time segments are highlighted which contain activities on the object. Selecting the time segments shows work processes with detailed information about the time spent with the different knowledge actions in the time segments. The filter functions and the neighbor exploration additionally help to recall the work process.

- **Object based associative search:** The information the subject strives for is not necessarily an understanding of a complete work process. Object based exploration can be used in associative search processes: identifying objects which were used together with a remembered object. The user enters an object name and identifies respective time segments in which knowledge actions were performed on the object. The identified time segments are selected and the searched object is displayed in the graph. The graph directly shows all other objects used in temporal proximity and shows those objects, the subject frequently switched to. Thus, associative searches like "the mail I read when writing the sales report" can be executed.

9.2.3.2 Time Based Exploration

Time based exploration refers to the use of the activity history to identify the activities performed within a specific time segment and to gain an understanding of the work process conducted in the selected period.

- **Time based work process recall:** A time segment is the entry point for a search of the work process performed in that time segment. By selecting a time segment in the timeline, a graph presentation of the time of interest is shown. The use of filtering, neighbor exploration and backward and forward move can help to understand the work process performed at the time of interest.

- **Time based associative search:** The subject does not necessarily want to understand a complete work process but wants to find an object which was used at a specific time while not remembering the name. Thus, the time segment is selected and the object is part of the visualized graph.

9.3 Recommendation: Activity-centric Recommender

In this section, the access of previously used information objects is supported by a proactive recommender approach named PASTREM[3]. This approach addresses memory threats related to the access of activity related elements and to activity switches.

Recommenders are generally used to help users to explore information collections under uncertainty. This is achieved based on rating the suitability of items for a user by identifying preference information [3]. Preference information results from observed activities (e.g., which products were looked at and which were bought in an online store). The reuse of information in information work can benefit from a similar approach. To address the uncertainty of actual information needs, the most recent activities are analyzed and help to identify the most recent information needs. Additionally, overall metrics like usage durations and usage counts of information objects are considered to identify information objects of general relevance.

A specific challenge is the weak structure of information work which includes times of very focused work and times of heavy multitasking. The recommender presented in the following addresses this challenge by being configurable for focused or multitasking oriented work.

The presentation follows the structure applied to the other methods. To classify the method in the design space, method design and interaction design are reported (see section 9.3.1). Then the recommendation process is reported (see section 9.3.2) and the approach is evaluated based on real work data (see section 9.3.3). Finally, a summarizing scenario is provided (see section 9.3.4).

9.3.1 Design Space: Method and Interaction Design

In the following the taken directions for method design and interaction design of the PASTREM recommender are reported.

[3] PASTREM refers to the supported process: the REMembering of useful information objects which already have been used in the PAST.

9.3.1.1 Method Design

The basic idea is to react dynamically on information needs which emerge during work processes. During an information worker's work process information needs occur. As a result, the subject frequently searches for information objects. If an activity is continued, the search and access activities are duplications of earlier search efforts because the subject already identified relevant information earlier. Proactive recommendation of relevant and already used information is the basic motivation for the PASTREM recommender.

The recommender mainly focuses on the supported access of activity related information objects (RQ2). Additionally, elements of general relevance should be recommended to help perform activity switches based on provided information objects (RQ5).

The following design direction with respective extraction tasks is valid for the PASTREM recommender:

- **Design direction:** The recommender direction is used for PASTREM. Recommendations are generated proactively and unobtrusively while the subject is working.

- **Extraction tasks:** The recommender supports the identification of relevant information objects based on their relatedness an activity (RO task, find by relation).

9.3.1.2 Interaction Design

In the following, the chosen interaction design for the PASTREM recommender is described:

- **Basic guidelines:**
  - *Simple, Easy to learn:* As a proactive recommender PASTREM is designed to automatically offer information. Thus, the recommender use is fairly simple.
  - *Permanent access:* The recommender is running permanently and generates recommendations.
  - *Efficient:* The efficiency of the recommender highly depends on the number of useful recommendations offered to the user. PASTREM is designed to provide recommendations for focused and multitasking oriented work situations. Based on this, the efficiency is intended to be increased.
  - *User acceptance:* The unobtrusive and yet useful access of information is relevant for user acceptance. This means that the user is not disturbed by recommendations but that he is able to check and access recommended information quickly. Notifications and overlay menus triggered by key combinations realize this.

- **Visualization choice:** A list visualization of information objects ordered by the likelihood has been chosen to visualize the recommendations. The amount of recommendations is limited to facilitate the identification of relevant information.

9.3.2 Process: PASTREM recommender

This section presents the PASTREM recommender approach. The PASTREM recommender builds on the CWO instance data created by the ContAct monitor and extends it (cf. section 6.4). The PASTREM recommender approach supports information reuse for information workers for a more focused or a more multitasking oriented work. The approach especially tackles the following aspects: 1) creating models for the recommender based on and within the actual work process, 2) limiting the required user input for the recommender system 3), structuring recommendation data in an easily accessible way to improve maintainability, and 4) respecting the dynamics of information work.

9.3.2.1 PASTREM Recommendation Continuum

PASTREM builds recommendations for information object reuse with respect to a work continuum which goes from an extremely focused, single task work to multitasking with frequent activity switches (see Figure 9.6). The assumption is that the actual useful recommendations differ. Focused work may be supported by information objects which are closely related to the task, even considering information objects which have been accessed very few times up until that moment. In contrast, a multitasking oriented work requires recommendations which support the activity switches by providing information objects as anchor points for upcoming tasks. An anchor point is an information object of high relevance which helps the user to quickly recall conditions and requirements of a task, like a memory cue that supports an activity switch. Therefore, multitasking oriented work would probably be supported best by information objects of general importance. Thus, the work continuum triggers a continuum of recommendations, focusing more or less on focused or multitasking work respectively.

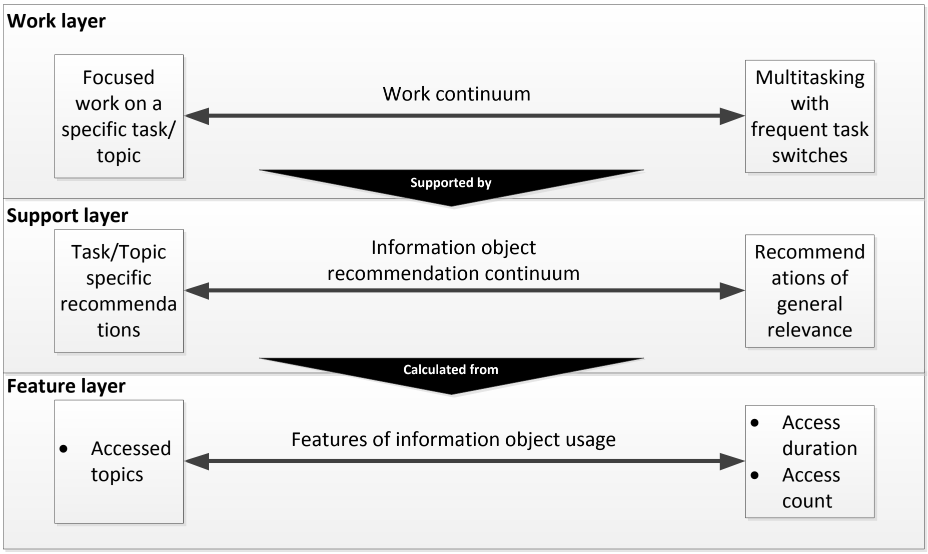

Figure 9.6.: Work continuum, related recommendation continuum and influence of features.

For PASTREM, three activity features are used: user topics, access count and access duration. Topics capture an abstract representation of information needs of the user generally related to the task a user works on. A latest time segment of user interaction is used to identify relevant topics which hint to related information objects in the interaction history of the user captured by the CWO. Topics can be understood as an information need following the assumption that a user continues to work on a focused task. Thus, topic related recommendations help users to focus on specific topics. Access count and overall access duration are global characteristics which are not related to the given focus task. Therefore, access count and access duration support activity switches as they result in information object recommendations of general high relevance, possibly unrelated to an active task but serving as memory cues for activity switches.

In the following, information about topic modeling and the integration of topics into the CWO is provided. Then, the overall process of PASTREM is presented, including data preparation and recommendation elicitation (see steps in Figure 9.7).

9.3.2.2 Topic Modeling for the Computer Work Ontology

Topic modeling stands for a group of approaches which use bayesian parameter estimation on multinomial distributions frequently used to derive the latent semantics of a text corpus. PASTREM uses the Latent Dirichlet Allocation (LDA) [27] to derive topics as latent semantics from a user interaction history as text corpus. In the following, a brief description of LDA is provided and the integration of topics extracted from interaction histories into the CWO is described.

The model assumption of LDA is that documents are composed of topics, while each topic is a set of words. Creating a document means choosing the required topics, their relevance for the document and sampling the words from the set of topics. LDA reverts this process and extracts a generative probabilistic model from a text corpus using Bayesian methods (for an introduction, see [115]). The model describes the probability that a word is part of a topic and the probability that a topic was used to generate a document.

Input for LDA is a bag of words representation of documents, i.e., the words used in the corpus are enumerated and for each document the count of each word is noted.

9.3.2.3 Putting Topics, Access Count and Access Duration into the Computer Work Ontology

The extended information object design pattern [95] describes the modeling of an information object. An information object can be realized by any sort of entity and can be about any sort of entity to express that a file has a content which stands for different topics. The following model applies: the file plays the role of abstract data and the abstract data expresses a topic which is modeled using the subject entity. As the topic extraction identifies a value which stands for the relatedness of the data to the topic, reification was applied.

An IO:SUBJECT gets connected to a CSO:MEASUREMENT unit with a property of type DNS:REFERENCES. The CSO:MEASUREMENTUNIT is again connected to an IO:INFORMATIONOBJECT. The measurement unit contains the relatedness value.

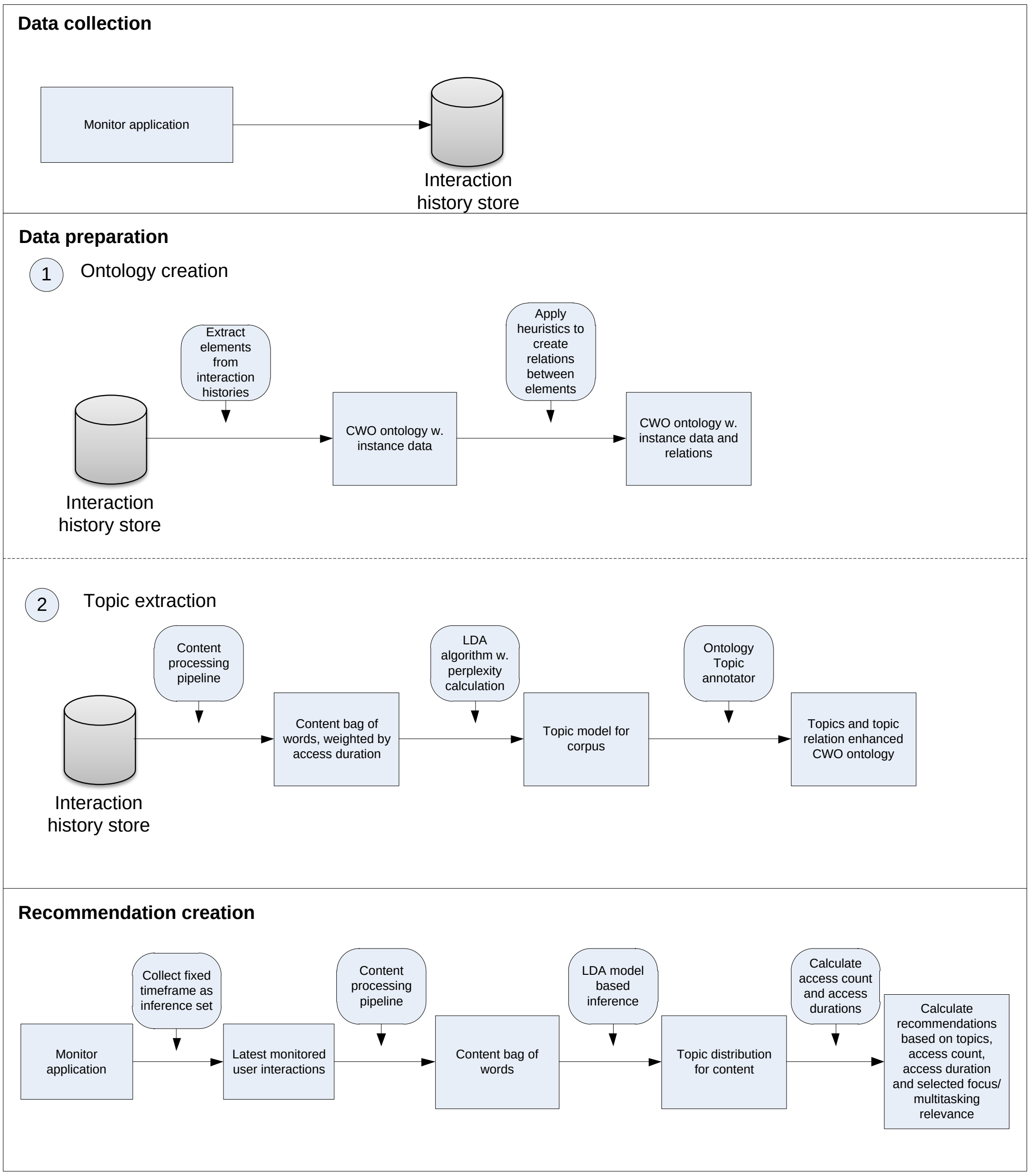

Figure 9.7.: Processes involved in the recommendation creation.

For access count and access duration, the extraction is simpler. They can be derived from the CWO based on the logged work situations which refer to information objects. The situation number for each information object needs to be counted to get the access count while the access duration is provided by the sum of the situation durations for each information object.

9.3.2.4 Data Preparation

The data preparation described in the following especially focuses on the extraction of topics from the interaction which requires the most effort within the recommendation process. Data preparation creates two artifacts which are used in the recommendation process. On the one hand, an instance of the CWO ontology is created and annotated with information about topics and the relatedness values for information objects. On the other hand, a model of the user topics is created, which is later used to infer topic distributions of new documents.

Data preparation is a time consuming task which needs to be performed on a regular basis (e.g., daily):

1. **Ontology creation:** First, the CWO ontology is filled with instance data about the elements the user interacts with. Based on the classification of information objects and additional heuristics, CWO instances are extracted. The resulting CWO ontology links information about the information objects, services and applications a user interacted with. The CWO also includes

information about work episodes, thus providing data about access count and access duration of the information objects. This is the output of the ContAct monitor.

2. **Topic model creation and ontology enrichment:** Second, the content of the interaction history is used to identify topics of the accessed content. This is done using LDA, which requires a bag of words representation of the content as input. The bag of words is created in a document processing pipeline, as it is frequently used in natural language processing tasks [193]. The pipeline contains the following elements: tokenizer, language detection based on n-grams, part of speech tagging and stopword detection. Stopwords are deleted and only nouns and verbs are processed further.[4]

   The pipeline creates content representations as bags of words: lists of words with the number of occurrences.

   The corpus represented by sets of bag of words is input to LDA. The LDA algorithm creates two distributions: a distribution of words to topics and a distribution of topics to documents.

   The ontology created in the previous step is enriched by the new data. Each topic is added as a topic entity represented by IO:Subject to the ontology. As described in the previous section, a CSO:MEASUREMENT unit connected with DNS:REALIZES connects CSO:ABSTRACTDATA played by the file and the IO:Subject.

   The output of the step is not only the ontology enriched with the topic and topic relatedness data. The second output is the model of document, word and topics created by the LDA algorithm which is used later for inference.

9.3.2.5 Recommendation Creation

Recommendations are proactively generated while the user is working. While access count and access duration are directly available, the relevant topics are derived from the latest interaction history. Therefore, the most recent segment of the user's interaction history is used as an inference set to identify the relevant topics.

The textual content of the interaction history fragment is used to identify recommendations based on the CWO ontology. To create recommendations, first a bag of word representation of the content is created using the document processing pipeline mentioned. The access date has no influence on the recommendation creation. The topic distribution for the content is inferred based on the model of document, word and topics created in the previous step. As a result a numerical representation of the topic relevance for the work in the considered latest time frame is created. The information object relevance ($IOTOPIC_{Rel}$) value is composed of the accumulated relatedness of the inference set to the topics and of the topics to the information objects: $IOTOPIC_{Rel} = (\sum_{t=1}^{T}(IS_t + \sum_{i=1}^{I} IO_{it}))$ with T=number of topics, I=number of information objects, IS_t= relatedness of Inference set to topic t, IO_{it} as relatedness of information object i to topic t. Thus, the relevance of a topic for the latest time segment adds to the relevance of all information objects for the topic.

For each information object, the relevance (IO_{Rel}) for the recommendation is calculated as a product of the topic relevance, the access count and the access duration weighted by factors to increase or decrease the relevance of focused or multitasking work respectively: $IO_{Rel} = IOTOPIC_{Rel}^{\beta} * ac^{\alpha} * ad^{\alpha}$ with ac as access count, ad as access duration in minutes and α and $beta$ to trigger the relevance of topics for focused work and of ac and ad for multitasking oriented work.

9.3.3 Evaluation

In the following, the PASTREM recommender is evaluated and compared to the results of other activity related recommenders: last recently used (LRU), semantic relatedness (TR), most often used (MOU) and longest used (LOU). LRU, MOU and LOU are self explaining. The TR algorithm recommends only based on the relatedness of the topic of the considered time segment to stored topic models with related information objects. MOU and LRU in particular are frequently used recommender types used in applications (often referred to as *recently used lists* or *histories*).

The evaluation is conducted in an ex post manner. Two interaction history data sets are used to identify the number of correct recommendations at a given position in the history by checking whether the elements actually accessed by the user would have been recommended. This results in a binary decision whether a used resource was recommended or not.

The evaluation process is described in the following. Information objects are identified which have been used in a real use time segment after a randomly selected starting point (see Figure 9.8, start point) in the interaction history and which were used earlier. The information objects of the real use time segment are compared to the recommendations generated by the recommender approaches, i.e., it is checked how many of the reused information objects in the real use slot are recommended by the algorithms (see Figure 9.8, use slots).

The events before the start position are used to create recommendations. Therefore, they are separated in two sets: 1) Model foundation set 2) Inference set. To ensure a sufficiently large number of events to build the model, it was enforced that the start position was at least in the "middle" of the interaction history. The recommendation inference set is a time segment of 10 minutes

[4] The natural language processing used for the recommender is similar to the process used for activity mining, cf. section 7.2.

before the selected position. This time segment is used for the recommendation creation. All events that occurred before than the recommendation inference set are used to build the ontology and to perform topic extraction (see Figure 9.8, model foundation and inference set).

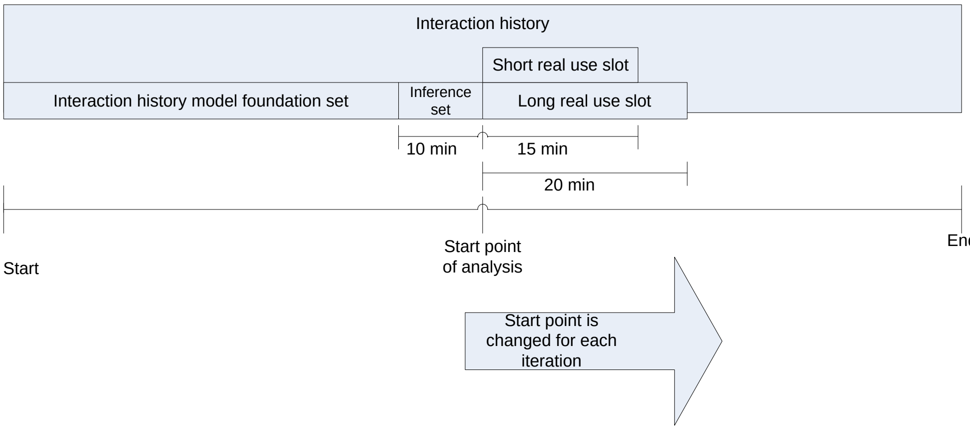

Figure 9.8.: Timeframes relevant for recommendation analysis for a given starting point.

9.3.3.1 Evaluation Method

The performance of PASTREM as well as the performance of LRU is scaled by the amount of elements included in the recommendation list. If both propose a list of all elements the user ever interacted with, both have the best possible recall but a low precision. This has practical relevance for the user interface of the recommender. A longer list of recommendations complicates user interactions due to limited cognitive capabilities. Therefore, the number of recommended elements is of high importance: the lower the number of recommendations required to make a valid recommendation, the better.

To address this, different recommendation set sizes have been compared: 10, 15, and 20 information objects. The ranking was performed as follows. For LRU the last n elements which were used directly before the beginning of the inference set have been used. MOU uses the n most often used elements and LOU uses those n information objects used for the longest amount of time. TR calculates the relatedness of the inference set to topics of the model and the relatedness of the topics to the information objects (actually the calculation of $IOTOPIC_{Rel}$ described in the previous section). Based on the resulting values, TR recommends the n elements with the highest relatedness. In all cases, elements from the inference set were excluded from the list of potential recommendations, as they are already used.

Another influence factor is the length of the real use slot. The longer the slot, the higher the probability that a recommendation might fit. This has been addressed by considering two different real use slot lengths: 15 and 20 minutes.

A third influence factor is the temporal length of the inference set. Based on experience, the length was set to 10 minutes. This value has not been changed in the study, although it is worth to investigate it further. The assumption is that the length of a useful inference time segment length depends on the homogeneity of work as measure for multitasking. An inhomogeneous work probably requires smaller inference time segments than homogeneous work.

Two interaction history data sets have been analyzed, using the described process. The α and β value were both set to one, to balance between task focus and multitask orientation.

9.3.3.2 Evaluation Setup

The interaction history data sets were created by researchers at an IT company. Data set1 contains 15363 interaction events (e.g., mouse clicks, window focus, etc.) for a period of 9 work days. Data set 2 contains 18311 interaction events for 4 work days. Information objects were only considered if they were focused at least 10 seconds. The data sets represent the normal working day of the two people (including normal activities like reading emails, browsing the internet, etc).

For data sets 1 100 data points and for data set 2 80 data points were chosen randomly with the constraint that at least one third of the overall event number was recorded before the selected event as starting point. The constraint assured that enough information objects and data for reasonable recommendations and topic model creation existed.

For data set 1 620 different information objects were accessed in all 100 real use time segments for a 15 minutes time segment (elements not included in the inference set). Of those 620 elements, 384 elements had not been used earlier, while 272 elements were reused. For all 20 minute real use slots, a total of 765 information objects were used, 436 had not been used before, while 329 were

	number of recommendations		
	10	15	20
PASTREM 15 minutes	42.6 %	58.1 %	67.2 %
PASTREM 20 minutes	35.6 %	39.2 %	68.1 %
LRU 15 minutes	41.5 %	42.2 %	49.6 %
LRU 20 minutes	40.1 %	41.3 %	49.2 %
MOU 15 minutes	43.7 %	64.7 %	69.1 %
MOU 20 minutes	43.2 %	64.7 %	69.3 %
LOU 15 minutes	24.2 %	37.5 %	54.0 %
LOU 20 minutes	24.3 %	37.1 %	54.7 %
TR 15 minutes	13.6 %	17.2 %	23.5 %
TR 20 minutes	12.7 %	16.5 %	22.4 %

Table 9.1.: Data Set1: Accuracy of recommendations for PASTREM, LRU, MOU, LOU, TR for a short (15 minutes) and longer (20 minutes) real use time segment of recommendation validity with lists of 10, 15 and 20 elements.

reused. The average number of reused information objects for a 15 minutes real use time segment was 2.7 and 3.2 for a 20 minutes real use time segment. Only three real use slots for 15 minutes as well as for 20 minutes reused more than 20 information objects which means that only for these three elements the largest recommendation set would be insufficient to recommend all items.

Data set 2 contained 287 different information objects accessed in all 80 real use time segments that had a length of 15 minutes. The 287 elements contained 237 elements not used before and 50 reused elements. Within the 20 minute time segments, 336 elements were accessed, 267 were unknown before and 69 were reused. An average number of 0.6 elements were reused within 15 minutes, 0.86 were reused within 20 minutes. No slot for 15 or 20 minutes contained more than 20 information objects, thus the recommendations could have been sufficient to recommend all actually used information objects.

The numbers already suggest different work styles captured by the data sets. In the following evaluation, one can see that data set 1 is more multitasking oriented while data set 2 stands for work with less multitasking which has effects on the different assessed recommender algorithms.

9.3.3.3 Evaluation Results

The accuracy of recommended information objects for PASTREM, LRU, MOU, LOU and TR for data set 1 is given in table 9.1 and for data set 2 in 9.2). PASTREM shows a good performance on both data sets, as up to 67.2 % and 71.0 % (15min) of accuracy is reached for a list of 20 recommendation elements and a 15 minutes time segment. For 10 elements 58.1 % (data set1), 54,7 % (data set2) and for 10 elements 42.6 % (data set 1), 40.4 % (data set 2) of all information objects used in a 15 minutes segment have been actually recommended.

An interesting result is the performance of MOU for data set 1 compared to the MOU performance for data set 2. While data set 1 reaches 69.3 % of accuracy a length of 20 minutes and 20 recommendations, data set 2 only shows an accuracy of 44.7 %. A similar peculiarity is the performance of LRU which shows a good performance on data set 2 reaching an accuracy of 63.6 % for 15 minutes and 20 recommendations while for data set 1 only 49.6 % of accuracy are reached for the same value. The overall weak performance of TR (23.5 % is the highest reached accuracy value) is another notable result. The different performances and especially the peculiarities with respect to the specific characteristics of the data sets are discussed in the following.

9.3.3.4 Evaluation Discussion

The evaluation showed a good performance of PASTREM for both data sets. The only algorithm with comparable results for data set 1 is MOU which shows a less favorable performance on data set 2.

Discussion of LOU and TR: LOU shows stable results between 24 and 50 % recommendation successes which show that the usage duration indicates relevance while it is not very useful on its own. The TR recommender shows exceptionally weak results. The assumption is that considering topic relatedness fails to rank the information objects which belong to the relevant topics. Additional relevance indicators are required to rank the information objects of one topic, e.g., frequently used for longer periods of time should be ranked higher than a resource which is only infrequently used for a short time. This is considered in PASTREM based on the integration of additional relevance factors which always influence the semantic relatedness based on an overall relevance (*ac* and *ad* are always bigger than 1).

PASTREM, MOU and LRU: A closer investigation of data set 1 showed a strong tendency of the user to switch between tasks. The good performance of MOU most likely results from the frequent activity switches which are best supported by recommending

	number of recommendations		
	10	15	20
PASTREM 15 minutes	40.4 %	54.7 %	71.0 %
PASTREM 20 minutes	36.0 %	47.5 %	59.6 %
LRU 15 minutes	29.5 %	47.7 %	63.6 %
LRU 20 minutes	25.3 %	44.4 %	60.3 %
MOU 15 minutes	31.7 %	41.5 %	44.7 %
MOU 20 minutes	28.3 %	38.3 %	40.3 %
LOU 15 minutes	30.0 %	40.0 %	48.0 %
LOU 20 minutes	27.5 %	37.7 %	44.9 %
TR 15 minutes	16.0 %	20.0 %	20.0 %
TR 20 minutes	14.5 %	18.8 %	18.8 %

Table 9.2.: Data Set2: Accuracy of recommendations for PASTREM, LRU, MOU, LOU, TR for a short (15 minutes) and longer (20 minutes) real use time segment of recommendation validity with lists of 10, 15 and 20 elements.

resources of an overall relevance without paying much attention to the topic which will change only minutes later. The second data set shows a more focused work type, even including phases of several minutes without any switch of the focus application. The good performance of LRU results from the stable work provided with data set 2 which creates strong local contexts of a high return probability to earlier used resources. For PASTREM, this data set benefits from topic specific recommendations ranked by access count and access duration.

Overall, the combination of semantic relatedness and relevance within PASTREM shows promising results. Next to the accuracy, the type of recommendations is of relevance. LRU and MOU tend to propose elements which were recently and often used, therefore, it is likely that the subject remembers those resources and the respective locations without help. In contrast, a review of the PASTREM recommendations showed that often elements not used for a longer period of time or with a medium access count (not the top4 and not the last4) were recommended. Those elements probably represent archived and ephemeral elements which are of specific benefit, as the recall of those elements is complex.

9.3.4 Summarizing Scenario

In the following an example scenario for the use of the PASTREM recommender during information work is provided. The example assumes that PASTREM runs while the user is working and that a slider can be used to configure the recommendations to address a more focused or a more multitasking oriented work. The slider is initially set to pure focused work.

A subject continues work on the preparation of a sales report. Therefore, the sales report document is most of the time in the focus. The subject already worked earlier on sales data which was tracked by the monitoring tool. As an effect of the topic based recommendation for focused work, recommendations will address sales and other information objects with a semantic similarity to the sales report. The subject wants to start working on a briefing document which was another activity very relevant lately. The subject moves the slider to multitasking. While the slider is moved the recommender list is dynamically modified. The amount of information objects semantically related to the sales report decreases. Information objects which have a higher general relevance due to their access duration and frequency appear. The briefing document appears in the list due to the duration and access frequency.

9.4 Design Cycle and Transparency 2.0

The support methods presented in the previous sections resulted from two iterations of the UCD-cycle. They have been realized in a prototype called Transparency 2.0. The result-oriented presentation in the previous sections neglected the first iteration which developed and evaluated a prototype named Transparency 1.0 based on the requirements discussed in section 5.4.[5]

This section addresses the first design iteration, initial design ideas (see section 9.4.1), their evaluation in the first prototype (see section 9.4.2) and the integration of those results in the second iteration (see section 9.4.3). It will become obvious that the relevance of technology acceptance and decoding effort for visualizations were underestimated in the first UCD cycle.

9.4.1 Characteristics of Transparency 1.0

The support methods included in Transparency 1.0 are closely related to the methods used in Transparency 2.0. In the following, the main differences between both are highlighted:

5 The user interface elements of Transparency 1.0 are provided in section E of the appendix.

- **Activity-centric task management:** The initial version of activity-centric task management allowed the organization of task elements based on mined activities. In spite of imitating well-known task management tools, the user interface focused on a simplified interaction with activity data resulting in an uncommon user interface design. A modal dialogue showed mined activities in a tree view and provided functionalities to merge existing task elements with mined activities and to transfer a bulk of mined activities directly in a row. Multiselects and deselects in the tree structure and in-place editing of names supported the process. The task management features as such were limited, as tasks only consisted of a name and an information object list. Due date, priority or classification were not implemented.

- **Interactive activity history:** The interactive activity history implemented a straight forward graph visualization: all knowledge actions identified by the system over time were used to produce one huge graph. Interaction with the structure became very complex due to the increasing size (cf. the earlier provided Figure 8.4).

- **Activity-centric recommendation:** Recommendations were only generated based on the mined activities. If an accessed information object belonged to a knowledge action in a mined activity, the other knowledge actions of the cluster were recommended. As a consequence, recommendations were only given for repeated knowledge actions. If an information object was not yet identified by the system, no recommendation was given. In contrast, PASTREM uses the content of unknown information objects and other characteristics of the history. Thus recommendations that consider the latest activities of the user are always provided.

9.4.2 Evaluation of Transparency 1.0

Transparency 1.0 was evaluated in a long term study which provided important information regarding user acceptance and method use.

For the evaluation of Transparency 1.0, nine users were recruited using convenience sampling. 7 were male, 2 female, with ages between 26 and 38. Users were either researchers or managers at an IT vendor and had significant IT experience. Their work included a high degree of self organization, involvement in multiple projects and commitment to an expert culture, thus fitting the profile of the information worker very well. None of them had used Transparency before. Users tested Transparency 1.0 for two weeks during their daily work activities (i.e., for 10 work days).

At the beginning of the study, they received a demonstration of Transparency's features and were asked to fill out one questionnaire regarding their personal working style and one regarding their impression of Transparency. They were asked to complete the latter again after the study had been completed. Additionally, an unstructured interview was conducted after the study. The interviews were evaluated based on clustering statements.

- **Activity-centric task management:** The participants did not consider the provided functionality as useful for task management. As 6 out of 9 participants already did task management they saw duplicated effort. To manage additional information like due dates, they used their standard task management technique and used Transparency additionally. In the final interviews it turned out that saving performed activities to keep track of the time spent and the improved quick access to information access were considered to be "unfinished features" by most participants. The participants saw it close to task management but they missed relevant functionalities.

- **Interactive activity history:** The evaluation showed that the cognitive effort required to understand the visualization and to filter required data from the visualization was not acceptable. Although the study participants initially appreciated the visualization, they were not able to extract the relevant information from the visualization. Further details about the respective results were already given in section 8.2.4.1.

- **Activity-centric recommendation:** The usefulness of the recommendation was not perceived by the participants. Some did not even recognize the existence of the feature during their daily work, as the data foundation accessible to provide recommendation was not big enough to identify enough recurring information objects to provide user support.

Overall, the participants failed to integrate Transparency 1.0 into their daily work processes. Therefore, they did not see a support of prospective or retrospective memory processes provided by the tool.

9.4.3 Resulting Effects on Requirements

The evaluation of Transparency 1.0 provided important information which had substantial effects on the context of use considered in the second UCD iteration. Although the initial context of use analysis based on activity systems was not wrong, two important aspects were not considered. To consider the gained information, the activity systems were modified in the following way:

- **Existing organization technique:** To address the double effort (maintain two organization techniques) to work on task execution the activity system 1 and system 2 which address the externalization of objectives and information access were modified (see section 5.4). A second organization technique was added to the tools and its maintenance was added to the workflow element.
- **Graph reading effort:** To address the effort of graph reading, a decoding likelihood rule was added to the systems 5 and 7 (see section 5.4). A tension between the rule and the objective emerges which complicates the outcome production and might result in the creation of new systems which only take care of visualization decoding.

Two new non-functional requirements were derived to address these tensions which were already mentioned at the beginning of this chapter:

- **RQ5: The system should be accepted by the user**
- **RQ6: The system interaction should require few cognitive resources**

The new requirements resulted in the modification of the activity-centric task management to follow existing designs more closely. To address RQ6, research on visualizations was conducted which is considered in the design space (see section 8.2.2) and resulted in different visualization choices for the task management as well as for the interactive activity history.

With respect to system 3, no modification was required. The provided first solution just proved to be too limited to actually remedy the effects of tensions 3 and 4. Therefore, a redesign was conducted and resulted in the PASTREM recommender presented in this chapter. Although the recommender still proposes data based on the history, the system combines two support methods and is able to provide recommendations even if users work on new, formerly unknown objects.

9.5 Evaluation

The design of the support methods integrated in Transparency and the underlying data collection mechanisms realize the requirements identified to address memory failures in information work. The design was especially modified to address the knowledge gained in the study of Transparency 1.0. This section evaluates the support methods by evaluating the integrated tool Transparency 2.0 with respect to user experience and the support of mnemonic processes. A controlled study was conducted to investigate memory effects by comparing prospective and retrospective memory of an experimental group of Transparency users and a control group of users who did not use Transparency. This evaluation will provide evidence that the support methods actually decrease the likelihood of prospective and retrospective memory failures in information work.

The evaluation is reported as follows. First, the hybrid study method which combines a questionnaire and interviews is described (see section 9.5.1). Second, details about the study setup are provided (see section 9.5.2). Finally, the results of the interviews (see section 9.5.3) and the questionnaire analysis are presented (see section 9.5.4).

9.5.1 Evaluation Method

To evaluate the support of prospective and retrospective memory based on Transparency 2.0 and the overall user experience and acceptance of the tool, a control study which combines an interview and a questionnaire for data collection was conducted. In the following, the study methodology is presented. The study is composed of two sessions for every participant with a duration of 1 hour each. The participants bring their own computer to both sessions. Altogether, 10 information workers participated in the study, resulting in an overall study duration of twenty hours.

- **First session:** The participants work on different tasks. This work is used to create a memory of executed activities to be recalled by the participants in the second session and to monitor the work processes for each participant Transparency 2.0 can be initialized with.

  For a participant, the first session is structured as follows. At the begin of the first session, Transparency 2.0 is installed and started on the computer of the participant. From the beginning, Transparency 2.0 logs all user interactions based on the ContAct component. During the first study, Transparency 2.0 is only used as a monitor, the participant does not interact with the user interface.

  Once the tool is started and runs in the background, the participant is asked to execute five information work tasks. The participant is asked to consider the tasks as real work tasks and to handle the information similar to the information he works with in real tasks, i.e., to use normal techniques to organize work items like folder structures. All participants followed this requirement.

  The tasks were designed to cover a variety of different characteristics. Four of the five tasks address topics the participants are familiar with (decide on applications, plan conference travel, identify project management numbers, create e-learning material) while one task addresses a domain which is new to the participants (compare different heating systems). Three tasks are communicated via email and two tasks are communicated orally. Tasks are given to the participants in random order. At random moments (approx. every four to eight minutes) new tasks are communicated to the participants and they are asked to switch to the new task directly, and to return to the previous task, once the new task has been completed.

All five tasks were given to all participants during the conducted sessions. Nevertheless, finalization of all tasks was neither possible nor intended within the sessions. If a participant seemed to be capable of finalizing all tasks, the study organizer closed the session early enough to assure that tasks remained unfinished. At least two tasks per participant remained unfinished to be memorized in the prospective memory.

Next to the data collected by Transparency, the study organizer made notes during the study, including details about work durations, sequences, interruptions and problems the participants faced.

- **Second session:** During the second study session the support of prospective and retrospective memory by current computer systems and especially by the Transparency 2.0 system was analyzed. Additionally, a user experience evaluation of the Transparency 2.0 system was conducted.

  The second session took place seven days after the first session (for one participant the study took place 8 days later). The participants were asked to recall the tasks and respective activities of the first session. The participants were divided in two groups with different recall support. An experimental group was allowed to use their computer with the data from the first session together with Transparency 2.0 which was initialized with the data collected during the first session. The computer and the Transparency tool were intended to support the recall process of the first group. A control group was only allowed to use their computer with the data from the first session to support the recall.

  The structure of the second session of the experimental group was as follows:

  1. *Introduction of Transparency 2.0:* The experimental group initially received an introduction to Transparency 2.0 on the computer of the study organizer. The organizer showed the main features (activity-centric task management and interactive activity history) of Transparency with demonstration data and the participants executed some small tasks to get used to the features.
  2. *Recall for prospective and retrospective memory:* The participants used their computers with Transparency 2.0. The tool provided access to activity clusters created from the data set of the first session and the logged activity history. In a semi-structured interview, the participants tried to recall retrospective and prospective aspects of the previous work session. The participants tried to recall the tasks together with details like accessed information objects, work sequence, interruptions and task durations. Next to these memories with retrospective characteristics, the participants were asked to recall which tasks were unfinished and how they would continue to work on these unfinished tasks to cover recall with prospective characteristics. The interview ended when the participants had recalled all details or stated that they could not remember anything else.
  3. *Questionnaire:* The participants filled out a Transparency related questionnaire. The questionnaire collected 1) information about different work and work organization characteristics for the participants, 2) the perceived usefulness of Transparency to solve the memory task 2) information about the usefulness of the interactive activity history and the activity-centric task management and 3) information about the usefulness and perceived ease of use of the whole Transparency system. The usefulness and the ease of use questions were taken from the respective questionnaire by [71] which is also used by Rattenbury to evaluate the CAAD system which shares some characteristics of Transparency 2.0 [218].

  The structure of the second session of the control group was as follows:

  1. *Recall for prospective and retrospective memory:* In a semi-structured interview, the participants were asked to recall the same information like the experimental group: tasks, task details, task sequence, interruptions, durations, status and open aspects of unfinished tasks. The participants were allowed to search for information on their computer. The interview ended once the participants were finished or did not recall any further details.
  2. *Introduction of Transparency 2.0:* The interview was followed by an introduction of Transparency 2.0 on the computer of the participants with the data collected in the participants' first session. The study organizer showed the main features of the tool and the participants solved a set of small tasks to get used to the features. Subsequently, the participants were asked to check the information they added to the system.
  3. *Questionnaire:* Like the experimental group, the control group completed a Transparency related questionnaire. The only difference was that the control group did not evaluate the perceived usefulness of Transparency during the recall process and instead was asked to estimate the usefulness of the tool for such recall processes.

The interviews for both groups followed a study guideline to explore details. The interview started with the open question "What was done during the last session?" followed by questions for more details, asking questions like "Was the task interrupted?" or "What exactly did you do to execute the task?". The interviewer directly wrote down the collected information and the participants were able to review and alter the answers during the study. The whole study was recorded on audio and the interviewer made notes during the session. The statements were clustered to identify those which were most frequently used by the participants. The collected data was compared to the data collected during the first study.

9.5.2 Evaluation Setup

Ten information workers participated in the study (9 male, 1 female). All participants worked as researchers in the research department of a large software vendor. The participants had 19 or more years of formal education. Most participants had more than 3 years of full time work experience (2 with 1-3 years, 6 with 3-5 years, 2 with more than 5 years).[6]

The perspective on work is very homogeneous among the participants. For most participants only some work tasks follow predefined structures (8 votes, 1 vote for no work task follows predefined structures, 1 vote for many work tasks follow predefined structures). The tasks themselves were described to consist of 50 percent new tasks and 50 percent recurring tasks (8 votes, 1 vote for "no work tasks are new", 1 vote for "many work tasks are new").

The workday is described as being roughly outlined in the morning but prone to modifications (9 votes, 1 vote "I know exactly how my day will look in the morning"). All participants stated that they work on more than two tasks per day (3 votes for "2-5 tasks", 4 votes for "more than 5 tasks", 2 votes for "Difficult to say. Many differences.").

While most participants have organization schemes for most information objects they interact with (9 votes for most information objects organized, 1 vote for all information objects organized), all state that they frequently search for information they have already saved or accessed with their computer (2 votes for "frequently search for information on the computer", 8 votes for "sometimes need to search for information").

The Microsoft Outlook Calendar and the scrum board are the most frequently listed tools to organize work execution. Some use task management systems, Post-it or physical notebooks. One participant sends himself emails to remind him of work tasks.

These answers show the mixture between work embedded in a predefined structure and knowledge-intensive individual work which was described as a characteristic of the information worker in the third chapter of this dissertation. All participants struggle with interruptions and need to react on unpredictable events on a daily basis. The interaction with information triggers duplication of search efforts despite the fact that the participants already maintain organization schemes for information.

9.5.3 Evaluation Result I: Interview

In the following an overview of the interviews conducted during the second study session is given. Based on an analysis of the relevant statements and the structure of the data collection process, different topics have been combined for the sake of reporting clarity. The data of the two groups is juxtaposed to show differences. Although some quantitative data is reported no further analysis of this data was conducted as the small sample size of five participants is too small for statistical treatment. Still, the homogeneity of the statements within the groups indicate that the collected qualitative data provides a valuable insight into the interaction of Transparency with memory processes.

- **Recall tasks and sequence:**
  - *Non-Transparency users:* The participants initially tried to recall without the help of the computer. Most participants were only able to recall 2-3 three tasks by memory. To validate the recalled information and to identify other tasks, a search activity on their computer desktop followed. Once interaction with the computer had begun, no participant ceased using the computer to support recall during the rest of the study. All participants applied a combination of email search and folder search. One participant additionally used the history of his web browser. For the emails, they filtered their mails based on the date and the sender. Most stated that this was a frequently applied strategy to "take a name or a date and get the respective email communication", as one participant explained. The folder structure generally followed an existing organization scheme. Some participants directly knew in which folder to look for data relevant for the study, others did not remember anything and started a brute force search within their folder structure. "I know that it is here, but I do not know exactly where" was the description of the search by one participant. A statement which is supported by the data collected in the questionnaire. Although all participants applied organization schemes, they still frequently search for information.

    Two participants failed to remember tasks they described to be the most challenging ones. For one this was screening applicants, for another one it was the task to collect information about a heating system. One participant did not recall the task to provide an internal project number which was a short task of 2 minutes duration, interrupting another activity. Overall, the recall of the tasks turned out to be very good, although it took participants between four and eight minutes until they had recalled every task. Most participants switched frequently between Outlook and the Windows Explorer during the recall process.

[6] The reported data was collected with the study questionnaire.

- *Transparency users:* Two participants directly started to use Transparency 2.0 and did not try to remember on their own. The other participants initially tried to remember and like the non-transparency users recalled 2-3 tasks and then started to use Transparency.

  The participants were not forced to use Transparency. Usage of other tools was explicitly allowed. Nevertheless, no participant left Transparency 2.0 during the course of the interview. Like the non-Transparency group, no participant of the experimental group returned to an unsupported recall process once interaction with the computer started.

  Three participants started to work with activity-centric task management and recalled their work based on the clusters they accessed. Those three participants later started to closely combine interaction with the history and the activity clusters. Two stated that they would like to have a closer integration to directly jump to the time fragments of an activity cluster and to have a visualization of the activity clusters within the interactive activity history. Two participants merely focused on the interaction history and chronologically browsed the logged interaction history and remembered based on the visible objects. One participant stated that "the history gives more control" and extensively used the filter feature. One initially transformed the accessed activity clusters directly into tasks during the interview.

  An interesting and critical aspect of the interaction was the immediate trust that was given to the Transparency tool. The participants seemed to rely completely on the visualized data without trying to recall much on their own. Once this behavior was recognized, the task clusters given to the participants were modified to miss the task which asked to provide information about a cost center and a project order, which was a short task. Out of three participants with this modification, two failed to remember the missing task. One remembered the task later within the study while browsing the history.

  The participants remembered all but two tasks which were removed from the activity clusters on purpose. The participants were able to recall the tasks organzied by Transparency very quickly and did not require additional support.

- *Comment:* Overall recall of tasks worked very well for both groups. Only the required effort differed, as the information is located directly in Transparency and no further application switches were necessary. Both groups showed that objects and even applications help participants to remember work tasks. One object which not necessarily contained detailed information about the task was sufficient for people to remember tasks. One example is a link "to the ISP system ... I used it to check something ... I checked for my internal project order number", which occurred for one participant. Another participant saw the Notepad application and remembered that he used it to structure details about applications.

• **Recall task status and sequence:**

- *Non-Transparency users:* The work on the sequence and the status was frequently combined. The sequence was recalled based on the emails and based on timestamps. The participants had problems to "place" those tasks which did not produce information objects and to recall the interruptions. Based on the sent emails and the artifacts all participants were able to remember the status of four to five tasks correctly. For some tasks there was uncertainty whether the product they created the week before would really be sufficient to finish the task. Overall, the recalled structures lacked information about interruptions. The random integration of the internal product order task which did not create an artifact was wrong in two of three cases.

  An interesting effect was a tendency to confirm recalled information. Even while the participants listed their tasks, they already talked about the status of those tasks. Later they would return to written emails and created information objects and open them to reassure themselves, "yes, as I already said, here is the list I constructed and here is the part I wanted to continue".

  A direct continuation of the work would have been complex for most participants, as they only checked information objects and emails they created but they did not remember which information sources they used to work on the information objects. Three stated that they would "google again". One stated that he tended to copy links to relevant sources into the information objects he produced but that he missed that during the study.

- *Transparency users:* All Transparency users used the history to answer the status question. Two participants stated that they would like to see "the products of work" highlighted in the cluster. The product of work was actually in the clusters based on the knowledge actions but it was not very prominent. Most combined the work on the status recall and the sequence recall.

  The sequence was recalled based on the interaction history by all participants. Two initially tried to recall the sequence based on the mined clusters but they missed "a visualization of the time I worked on it", as one of them stated. Three participants stated that it is "confusing to find the interruptions", as the time segments do not offer a further linearization of the work process. Still, all but one participant were able to give a precise description of their work process, including each interruption. When the sequence was recalled, the correct status was recalled, too.

  The participants showed certainty to be able to continue work on the respective tasks. Based on the duration information and their own recall, they remembered the relevant information sources they had used to work on and would have opened those directly to continue working.

– *Comment:* Sequence recall was generally difficult. Without Transparency, participants were forced to rebuild the sequence based on date and time information attached to information objects, esp. emails. With Transparency the sequence can be accessed directly based on the history. Nevertheless, the interaction with the history still requires substantial effort. One participant stated that for daily work a sequence recall was not very relevant for him, as only the priority counts.

  The status recall with respect to relevant information was difficult without Transparency. Transparency helps to directly know which information is required to continue working. Additional suggestions were made by the participants on how to further improve the support.

- **Recall Duration:**
  – *Non-Transparency users:* Only one participant was able to make correct ad-hoc estimations for all tasks. Two participants made good estimations for two or three tasks and bad estimations for the other tasks. Two participants made incorrect estimations for at least four of the five tasks. This shows the complexity of recalling the time spent with an activity. Two participants explained that the frequent interruptions made it very complex for them to remember the actual durations of the activities. The three participants with good estimations for two to three tasks identified durations by taking the duration of the first meeting and step by step optimizing time distributions based on the task complexity. Although the process worked for the study, it would not be applicable to real work scenarios.

    When one participant later reviewed his data in Transparency that he "was not aware that I wasted so much time with work on screenshots" when he created the e-learning material. He added later "generally I really would like to know how much time I waste with unimportant activities during the day".

  – *Transparency users:* Four of the five Transparency users used the time provided by the clustered activities. One participant rebuilt it based on the interaction history. The identified times were good estimations compared to the manual transcript made during session 1. At the same time the participants showed a high trust to the data of the tool and no participant stated that he was skeptical about the time provided by the tool.

  – *Comment:* Duration turned out to be the very complex without Transparency. An effect already reported by Bellotti [24]. Here, Transparency turned out to be very useful.

- **Recall detailed work process for one task:**
  – *Non-Transparency users:* The participants provided a high level overview of the work tasks which focused on the topic of the task, the artifact they tried to produce and the decisions they took. In the terminology of this dissertation, the participants renarrated knowledge actions, but they lacked information about the information objects attached to the knowledge actions: "I accessed a website with information about trains from Edinburgh to St Andrews" or "I copied information from a website about heating systems to the excel sheet." If they used a service they accessed frequently (e.g., the hotel booking portal or a website to compare flight prices), they also named information objects, but most descriptions lacked information about the name or address of respective information objects. Sometimes participants even missed the most central information: For the conference planning task, a participant did not talk about the source of the information about the conference, the conference website, although he considered this information to be important. When he later saw Transparency 2.0 with his data displayed in the history, he directly said "ah, the conference website, I absolutely forgot that I used that". An interesting effect was the recall of queries. All participants remembered the search terms they used to identify information but they were not able to recall details about the information objects they accessed based on the search terms (e.g., "something about heating").

  – *Transparency users:* All but one participant used the interactive activity history to provide task details. The participants were directly able to narrate the task in a very detailed manner and offered deep insights into their work process by combining high level information about reason of interaction with detailed information about the information objects and the browsing paths they operated on. Two also gave brief details to explain the switching frequency between different objects. One participant cross-checked his narration based on the history with the hierarchical compound graph of the provided cluster. One participant used the statistic view of the cluster to describe relevant resources and was able to combine high level information with detailed resource information, too. This participant showed explicit interest in the queries he conducted and said "the queries are very important to me", supporting the observations for query recall of the non-Transparency user group.

  – *Comment:* The detailed work process recall shows the strength of Transparency with respect to the support of detailed information about the work processes. Transparency provides the details which would be necessary to continue the work and to recall relevant reasons which lead to decisions. Without Transparency, the recall process focuses on topics and work process types.

When non-transparency users first were introduced to the tool and started interacting, there was a general appreciation of the tool and an immediate understanding of the relevance for the recall task they had just performed. Statements given were, "wow, this would have been so much simpler", as one participant stated. One participant who began to work with Transparency even enjoyed crawling his history, stating "I think I would check this data all the time."

The perception of the usefulness for the recall task was considered similarly positive by the participants of the experimental group and by the participants of the control group (see Figure 9.9). Both vote for an average of 2 on a Likert scale from -3 to $+3$ with a std of 1,18.

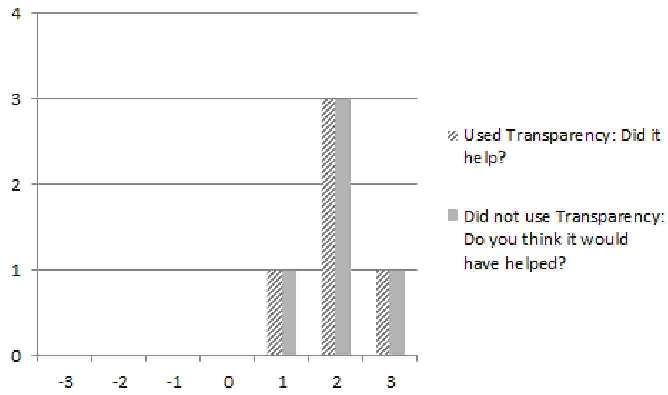

Figure 9.9.: Transparency usefulness: experimental group and control group.

Overall, all participants considered Transparency to be helpful for execution of recall tasks. Additionally, some peculiarities can be derived from the described interviews:

- **Transparency 2.0 supports retrospective memory:** Transparency 2.0 simplifies recall, as required information is encapsulated in one tool. At the same time, Transparency users showed a higher correctness and a higher degree of detail of recalled information especially for duration, sequence and status information. The positive comments of the non-Transparency users when they first used Transparency add to this positive impression. Non-Transparency users were additionally able to enhance their earlier report and stressed the positive effect of Transparency on recall processes. The study shows that only one information object representing a file or even an application helps people to rebuild the events around this object.

- **Transparency 2.0 supports prospective memory:** The recall of task status and the direct association of the status with the relevant information to continue the task in contrast to the "google again" statement of the control group indicates that Transparency supports prospective memory. Nevertheless, a deeper investigation into the actual integration with the task management was not covered in this study. One single information object displayed to the user was sufficient to recall a task status and a possible open activity. Thus, a displayed information object not only supports retrospective but also prospective memory.

- **Transparency 2.0 affects transactive memory:** Transactive memory is a form of remembering how to access information. This kind of memory releases the burden of memorizing everything. In spite of memorizing a complex fact one can also memorize who in the social network has the respective knowledge or where the information is situated. Transactive memory has been recently discussed with respect to the internet and especially Google [265]. The immediate reliance on the data provided by Transparency indicates that people would cease to rely on their own memory to memorize and recall facts about their work but would begin to delegate this part of the memory to a tool like Transparency 2.0.

9.5.4 Evaluation Result II: Questionnaire

In the following, the assessment of the main features of Transparency 2.0, the interactive activity history and the activity-centric task management collected by the questionnaire is provided. The results of the experimental and the control groups are combined, as both groups spent a similar amount of time with the tool during the second session, one time before and during the interview (experimental group) and one time directly after the interview (control group).

- **Overall tool perception** The detailed results for usefulness and ease of use are given in table 9.3. The results for the average perceived usefulness responses (mean 1.7, std 0.81) and for the average perceived ease of use (mean 1.72, std 0.97) on Likert scales from -3 to $+3$ show an overall positive perception of Transparency with respect to usefulness and ease of use. For the CAAD tool the same questions were used to assess usefulness and ease of use [218]. Although a direct comparison of the values is not advised due to a different evaluation scenario, the similarity of the tools proposes a comparison as an indication.

Question	Mean	S.D.	
I expect that using Transparency for my work would help me to accomplish tasks more quickly.	1.4	0.96	**Perceived usefulness**
I expect that using Transparency would improve my work performance.	1.8	0.63	
I expect that using Transparency would make it easier to do my job.	1.3	0.82	
I expect that using Transparency would enhance my effectiveness at work.	1.9	0.74	
I expect that using Transparency while I work would increase my productivity.	1.5	0.85	
I would find Transparency useful in my work.	2.3	0.48	
Average perceived usefulness score:	**1.7**	**0.81**	
I find it easy to get Transparency to do what I want it to do.	1.4	0.97	**Perceived ease of use**
My interaction with Transparency is clear and understandable.	1.3	1.16	
Learning to operate Transparency was/is easy for me.	1.7	0.95	
It was/would be easy for me to become skillful at using Transparency.	2.1	0.88	
I find Transparency easy to use.	2.1	0.74	
Average perceived ease of use score:	**1.72**	**0.97**	

Table 9.3.: Questionnaire results. Questions were scored on a 7-point Likert scale ranging from -3 to $+3$. Mean and standard deviation response values are reported for each question and for each overall response average.

Transparency performs significantly better for ease of use (T-stat=-2.5 p<=0.03) and usefulness (T-stat=-2.8,p<=0.02). This can be considered as an indication that the close alignment with known tools: task management and histories helps users to understand the benefits of applications that use historic data more intuitively.

- **Details in activity-centric task management:** The detailed results for the interactive activity history are given in table 9.4. The benefit compared to normal task management is clearly perceived by the participants (mean 2.4, std 0.52). One possible reason is the simplified maintenance of tasks (mean 1.7, std 0.87) and additional features highlighted by the participants of the interview: time tracking, statistics and graph based work process visualization.

  One design focus of activity-centric task management was prospective memory "what still needs to be done". The participants see support for this (mean 1.4, std 0.97), but as indicated by the high standard deviation next to very positive perceptions of this aspect, many neutral answers exist. One reason for this might be the focus of the study: although participants were asked to remember task status and the means of continuing tasks, they did not really work on prospective processes based on activity-centric task management. The support of activity switches is clearly recognized by all participants (mean 2.1, std 0.56).

  The strong value for the assessment of activity switching support shows that the features clearly support this activity. Still, the activity switching is mainly associated with activity-centric task management (mean 2.1, std 0.56) while the interactive activity history is less strong (mean 1.2, std 1.23).

- **Details on interactive activity history:** The detailed results for the interactive activity history are given in table 9.4. The strong mean values for "access to data not accessible with other programs" (mean 2.3, std 0.81) shows the perceived novelty of the approach. The aspect "gives insight into work I did not have before" (mean 1.9, std 1.6) with the high deviation showing that many participants strongly assume an increasing awareness while others have a neutral opinion about the gained insight into work.

  The participants strongly associate the support of retrospective memory with the interactive activity history as the strong value for "helps to remember what was done" (mean 2.3, std 0.82) against a weak value "remember what needs to be done" (mean 0.1, std 1.67). At the same time, the interview showed that some participants remembered details about the tasks they wanted to continue when they used the history. Still, the personal evaluation shows that this functionality is not perceived by the participants.

Question	Mean	S.D.	
Activity-centric task management helps me to remember what was done.	2.2	0.63	Activity-centric task management
Activity-centric task management helps me to remember what still needs to be done.	1.4	1.4	
Activity-centric task management simplifies switching between different tasks.	2.1	0.56	
Activity-centric task management simplifies creation and maintenance of tasks (compared to other task management tools.	1.7	0.87	
Activity-centric task management has benefits compared to normal task management.	2.4	0.52	
Interaction with Activity-centric task management gives insight into my work I did not have before.	2	0.81	
Interactive activity history helps me to remember what was done.	2.3	0.82	Interactive activity history
Interactive activity history helps me to remember what still needs to be done.	0.1	1.67	
Interactive activity history simplifies switching between different tasks.	1.2	1.23	
Interactive activity history gives access to data not accessible with existing programs.	2.1	1.5	
Interactive interaction history gives insight into my work I did not have before.	1.9	1.6	

Table 9.4.: Questionnaire results. Questions were scored on a 7- point Likert scale ranging from -3 to $+3$. Mean and standard deviation response values are reported for each question and for each overall response average.

9.5.5 Intermediate Results

The evaluation reported in this section has provided indication that the support methods activity-centric task management and interactive activity history decrease the likelihood of prospective and retrospective memory failures compared to the use of standard applications.

The results show that the activity-centric task management especially has benefits with respect to prospective memory failures while the interactive activity history mainly addresses retrospective memory failures.

An unanticipated result is the effect of the tool on the transactive memory. The participants relied on the tool without crosschecking results based on their memory. Thus, the tools not necessarily mediate the recall process but replace the cognitive effort involved in the recall process almost completely.

9.6 Summary

To address memory threats within information work, three user support methods have been presented and evaluated. An evaluation of the methods has shown that the methods decrease the likelihood of memory failures. Thus, it has been shown that activity-centric methods of information work support are a valid answer to the research question *How to support mnemonic processes involved in information work at the computer workplace?*

The methods follow the different design directions:

- **Activity-centric task management:** The creation and maintenance of task lists is facilitated and enriched by work process information. The method supports the organization of work processes to recall activities and simplify activity switches. Questions like “What do I still have to do?”, “How much time did I work on writing the sales report” or “What do i need to continue working on the sales report” are answered.

- **Interactive activity history:** The time and object based exploration of work histories is enabled and facilitated by search and filter methods. The method answers questions like “Which websites did I visit when I worked on the sales report?” or “What did I do yesterday morning?”

- **PASTREM recommender:** Recommendations are proactively generated based on the accessed information. This is intended to address emerging information needs and to simplify the access to the respective information objects. The method supports focused or multitasking oriented work types.

The section has shown the usefulness of the design space to develop and classify support methods.

The support methods resulted from two iterations of the UCD cycle. The first cycle resulted in a prototype named Transparency 1.0. The evaluation of Transparency 1.0 especially showed the relevance of technology acceptance and the need to consider the complexity of visualizations with respect to the decoding of the required information to solve the task. These two aspects resulted in substantial enhancements of the user support methods when Transparency 2.0 was developed.

The evaluation of the PASTREM recommender has shown that 50 % of all reused elements for a 10 item list and 70 % for a 20 item list were actually recommended. These results show that the recommender is capable of supporting activity switches and focused work processes by proactively delivering information. An evaluation of the activity-centric recommender has shown that at least 67 % of the actually reused information objects within a twenty minute timeframe were recommended by the approach. The recommender performed better than related approaches. For the interactive activity history and the activity-centric task management system, the actual support of mnemonic processes was shown. Both methods were successfully applied to recall information about task execution processes performed one week earlier. For the first interaction with Transparency 2.0 within the user study which had a focus on the use of activity-centric task management and the interactive activity history, a promising ease-of-use score (avg. mean 1.7, avg. std 0.81) and a promising usefulness score (avg. mean 1.72, avg. std 0.97) show an overall appreciation of the Transparency 2.0 tool (used questionnaire: [71]).

10 Conclusion

The main topic of this dissertation is the reduction of memory failures in information work at the computer workplace using a design science approach. To design a system to address memory failures, information work has been analyzed using user-centered design (UCD) with the activity theory based system design method (AT-SDM). As a result different support methods to provide memory cues were created. All support methods build on externalized work data in terms of interaction histories.

In the following a summary of this dissertation is provided. Within the summary the contributions of this dissertation are highlighted (in *italics*):

- **Part I—Analysis of the information work process (exploration based on literature and study):** Chapters two and three have provided *theories and background information on work* in general and information work in particular used throughout the whole dissertation. Work has been discussed in terms of psychology, organization theory and sociology as a foundation for the characterization of information work. Those characteristics relevant for this dissertation have been used to create an *information work ideal type* (see section 3.1). The ideal type characterizes information work as work governed by multiple underspecified objectives subject to various constraints to be tackled in an environment composed of a variety of tools to access, create, modify and disseminate large amounts of weakly structured information. External interruptions and self-interruptions are caused by and result in dynamic re-prioritizations of objectives and create a work governed by multitasking and frequent activity switches. Prospective and retrospective memory failures result from the cognitive challenges of the complex and dynamic information work process.

  Despite the focus on the unpredictability of information work, the ideal type covers recurring activities in information work on the level of actions and (possibly) operations. These aspects are covered in a taxonomy of knowledge actions and desktop operations for information work at the computer workplace. The taxonomy was identified in an explorative study (see section 3.3) and has been created in close alignment with Hädrich's work on knowledge actions and activity-theory (AT).

  The overall conclusion of the ideal type: information workers frequently face new and underspecified objectives which are realized in activities which reuse and adapt well-trained work techniques (knowledge actions and desktop operations). Thus, the underspecified and new activity is addressed by the well-known, the known work techniques.

- **Part II—User-centered design based on tension analysis (design method):** Chapters four and five focus on system design methods in general and on their particular application to the domain of information work at the computer workplace. The identification of requirements for a tool to actually support information work was the consequent next step, once the environmental factors which trigger prospective and retrospective memory failures were identified. Despite the detailed information gathered for information work, the requirement identification is no straight forward process. There is no obvious way to transfer characteristics of information work like the relevance of cognitive processes to coordinate multiple goals (while having high degrees of freedom) into a context of use model.

  To address the problem of modeling information work in a framework for system design, a method set for the UCD process was proposed: *AT-SDM* (see chapter 4). AT-SDM provides methods to be applied in the first two steps of the UCD cycle, the context of use analysis and the requirement elicitation. The process is as follows: the AT-SDM generates activity system models (ASMs) which stand for (possibly related) activities. Based on relations, parallel activities and activity switches can be modeled. Each ASM can be analyzed with respect to inter- and intra-model tensions which suggest problems which may complicate the successful execution of a work task. Based on the tension analysis and a model transformation process to alleviate tensions, the context of use is directly converted within the AT-SDM into a set of requirements.

  Based on UCD with AT-SDM, the information work process was analyzed with a focus on tensions related to memory failures (see chapter 5). *Six tensions were identified* that rely heavily on mnemonics: *Forget tasks (inter-model), Forget task status (intra-model), Maintain active activity system (intra-model), Maintain multiple active activity systems (inter-model), Separating knowledge actions (inter-model), Interruption (inter-model)* (for details, see table 5.1).

  To alleviate the tensions, the ASMs were transformed. The transformation was directed towards the initial idea of providing memory cues. To implement the transformation, requirements were identified. The requirements elicitation was informed by a detailed state-of-the-art analysis of tools to support information work based on information externalization. In the following, *the derived requirements* are summarized (for details, see section 5.2):

  - **RQ1:** The system should help derive existing activities (Tension T1). Based on system 1.
  - **RQ2:** The system should help derive activity related elements (Tension T2, T3, T4). Based on system 2.
  - **RQ3:** The system should help derive connections of activities (Tension T5). Based on system 3.

- **RQ4:** The system should help derive executed work processes (Tension T2, T3, T4). Based on system 4.
- **RQ5:** The system should support activity switches as identifying and activating an activity system (Tension T6). Based on system 5.
- **RQ6:** The system should use data about the information worker's work process (required by RQ4), existing activities (required by RQ1, RQ5), connections between activities (required by RQ3) and the involved elements (required by RQ2).
- **RQ7:** The data should be collected unobtrusively and require little maintenance effort by the user.
- **NF-RQ1:** The use of the system should be simple, easy to learn and quick.
- **NF-RQ2:** The system needs to be seamlessly integrated into the computer system of the user to be accessible during each activity.
- **NF-RQ3:** The system needs to operate efficiently to have a good user experience as it will run permanently.
- **NF-RQ4:** The system should protect the privacy of its users by preventing misuse of the activity data.
- **NF-RQ5:** The system should be accepted by the user.
- **NF-RQ6:** Information extraction from the system should be simple.

- **Part III-1—Information work process elicitation:** The chapters six and seven address work data externalization with respect to work data collection, work data processing, work data formalization and activity mining. The work on work data externalization addresses the requirements (RQ6-RQ7) and considers the privacy constraint by the proposed architecture (NF-RQ4).

  The ContAct monitor has been designed and implemented as an integrated application to take care of work data externalization. The monitor implements a *process of interaction data management* including data collection, data processing and data organization. For the data collection, the ContAct monitor includes a variety of software sensors to capture events of user system interactions in an interaction history. Specific attention has been given to texts and information objects: ContAct captures which information objects were accessed and which content was visible on the screen. The data processing covers *a rule and heuristic based extraction of desktop operations and knowledge actions* based on the logged interaction histories. The collected and derived data is formalized using the *computer work ontology* (CWO). CWO formalizes the hierarchical model of activities, knowledge actions and desktop operations and connects those work activities to a formal model of the computer workplace (e.g., an authoring knowledge action performed on an application for a work scenario and modifying a file with a location and content). A work process turns out to be a graph of knowledge actions.

  To address the multitasking driven work execution, research on *activity mining for information work* at the computer workplace was conducted. As work operations for different activities might occur in close temporal proximity the identification of activities is a complex problem. The lack of knowledge about existing tasks and respective activities complicates activity mining further. In this dissertation activity mining is tackled as a clustering problem: the graph of knowledge actions representing the work process is the input to identify a set of clusters within the graph standing for activities. Three approaches were considered. 1) Semantic approaches make use of the semantic relatedness of knowledge actions in the clustering process. 2) Process approaches cluster knowledge actions based on the switches between knowledge actions. 3) Hybrid approaches combine different characteristics of the knowledge action graph to realize the clustering process.

  The considered activity mining approaches have been evaluated twofold. A first set of methods was tested against a gold standard. Those methods which performed best were used to cluster five days of real work data which was manually evaluated by the person who generated the data.

  The evaluations have shown that semantic similarity is an important factor for the identification of activities. The best results for the gold standard were achieved with a hybrid approach (Precision 0.72, Recall 0.728, F-measure 0.72), combining semantic similarity and graph topology. For the real work data the purity of the clusters was clearly increased for a pure VSM approach (mean 1.9, std 1,8 for VSM, mean 1.3, std 2,4 for hybrid on a 7-point Likert scale from –3 to +3) while the recall of work and the perceived ability to continue work increased for the hybrid approach (mean 1.3, std 2.1 for VSM, mean 1.5, std 2.2 for the hybrid approach on a 7-point Likert scale from −3 to +3). The graph topology approach using LinLog showed interesting results for the gold standard (Precision 0.615, Recall 0.615, F-measure 0.615) but did not perform well for the real work data (purity of mean −0.05, 2.6 std on 7-point Likert scale).

  Overall, the results are promising and indicate that activity mining on the computer desktop can generate results which actually help to understand information work and support it. Nevertheless, additional investigation into the domain is required, especially with respect to the combination of different indicators within the activity mining process (for details, see the related discussion in section 7.5).

- **Part III-2—Support method design and showcase:** Chapter eight contributes a *design space for user support methods to address memory failures*. The design space helps to develop support methods which address the identified requirements by specifying design directions (exploration, organization, recommendation) and interaction techniques to interact with activity data. *For each design direction one user support method has been created*. The support methods are the result of two UCD process iterations. The methods created during the first UCD-cycle showed problems with respect to the integration of the methods into the work process and with respect to the access of data by the user. Based on the improved domain understanding provided by the evaluation at the end of the first design iteration the context of use model was adapted in the second iteration of UCD. The methods which were created during the second iteration are described in the following. The methods created in the first iteration have been implemented prototypically in an application named Transparency 1.0 (for the discussion, see section 9.4), those developed in the second iteration have been implemented in Transparency 2.0.

The Transparency 2.0 application uses the data collected by the ContAct monitor to realize the following support methods:

– *Activity-centric task management:* To support the organization of the personal work a task management system is provided. Activity data enriches the task management system by 1) simplifying the creation and maintenance of tasks 2) increasing execution process awareness and simplifying activity switches based on activity data attached to tasks (see section 9.1). The main data source for the activity-centric task management are mined activities.

  Task objects include information objects relevant for the task, temporal information (task activation times) and a compound graph visualization of the work process. For each task object, the time spent with the information objects attached to the task as well as the work process information is constantly updated. Additionally, new information objects are proposed based on new mined activities related to an existing task object.

  The task management was evaluated in a user study (see section 9.5). The participants acknowledged retrospective memory support (mean 2.2, std 0.63 on a 7 point Likert scale from -3 to $+3$). With respect to prospective memory different opinions emerged. The result was a mean of 1.4 with a standard deviation of 1.4, thus showing that next to many positive opinions (maximum support of +3) many neutral opinions (vote for zero) also existed. All participants supported the perspective that the activity-centric task management has benefits compared to existing task management approaches (mean 2.4, std 0.54) and that the creation and maintenance of tasks is simplified (mean 1.7, std 0.87). The participants also agreed that switching between different tasks is supported (mean 2.1, std 0.56).

– *Interactive activity history:* The interactive activity history allows the exploration of the knowledge actions of the information worker over time (see section 9.2). The user interacts with a timeline to access knowledge actions activated during selected time segments. Additional functionalities are keyword search, filtering, neighbor exploration and the dynamic visualization of time. In particular, the keyword based search is an interesting way to explore the earlier user interactions as for a keyword, all time segments which used an information object related to keyword are highlighted. The interactive activity history thus supports questions like “what did I do yesterday morning?” by browsing the timeline and associative search like “when I worked on a mail to my boss I read an interesting article, how can I find it?” by searching for the remembered object and following the graph edges. Another supported question is “when did I work on documents about mechanical engineering?” by searching for keywords.

  A user study showed that the history helps people to remember rich details about their performed activities (for details, see section 9.5). The questionnaire results show that study participants approve that the history helps to remember what was done (mean 2.3, std 0.81 on a 7 point Likert scale from -3 to $+3$) . Support for prospective memory was not seen by the study participants (mean 0.1, std 0.87). Participants did not have shared opinions whether the history provides new insight into their work (mean 1.9, std 1.6) and whether the history provides new information (mean 2.1, std 1.5). The high standard deviation for *information not accessible with existing programs* and *gives new insight into work* shows that the opinions, in fact, have two biases, neutrality (0) and overall approval ($+3$).

– *Activity-centric recommender:* While the user is working, information objects from the user’s interaction history are proactively generated. The recommender can be configured to recommend for more multitasking or for more focused work (see section 9.3). For a more multitasking oriented work, the recommender considers the usage frequency and the usage duration of all information objects the information worker interacted with. For more focused work, the algorithm identifies relevant topics within the latest knowledge actions applied by an information worker. The topics are taken from a topic model based on the complete interaction history of the user. Based on a calculation of relevance, information objects are identified which are closely related to the latest relevant topics of the information worker.

  A balanced configuration of the algorithm (equal relevance of topic and earlier access count/access duration) was evaluated on two long-term data sets of information work. PASTREM proposed at least 67 % of the actually reused information objects within a twenty minute timeframe and performed better than compared approaches like most frequently used, last recently used and topic based.

Transparency 2.0 was evaluated in a user study which had a focus on the use of activity-centric task management and the interactive activity history. The study resulted in a promising ease-of-use score (avg. mean 1.7, avg. std 0.81) and a promising usefulness score (avg. mean 1.72, avg. std 0.97) which show an overall appreciation of the Transparency 2.0 tool (used questionnaire: [71]).

10.1 Approach Discussion

This dissertation has used activity data to support information workers and reduce prospective and retrospective memory failures. Three aspects of this solution seem to be of specific relevance for upcoming research on information work and respective support. These aspects are summarized in the following.

10.1.1 Information Work

The described information work ideal type challenges simplified perspectives on information work by considering challenges like multitasking, activity switches and underspecification of work items which have effects on information work execution. Therefore, the ideal type helps to design applications for information workers without missing important characteristics which might result in a rejection of a new technology (for an example of failed introduction, see [249]). The AT-SDM with an analysis of inter- and intra-model tensions of ASMs helps to actually consider important aspects of the ideal type within the system design process.

10.1.2 Characteristics of Activity-centric Systems

This dissertation can be understood as a contribution to the domain of activity-centric applications. Rattenbury finishes his dissertation about “An activity based approach to context aware computing” with the hope to “inspire more research attention on technology that can effectively handle and respond to the nature of human activity and its relationship to context”. This dissertation follows Rattenbury. Here, the direction mainly focuses on a generic, yet important aspect, information worker’s memory for multitasking-driven underspecified information work.

The promising results show that the use of activity data to offer user support is a relevant direction. The use of such data is not new, in fact, activity data is used in many support mechanisms. Recently used file lists, autocomplete fields based on earlier entries and histories are some examples. Still, the existing use of activity data tends to be application-specific recommendations (e.g., within the word processor the last accessed documents are shown). By focusing on activity data as a data type on its own, mechanisms of application independent capturing and services to collect and disseminate activity data to consuming applications would provide a valuable resource for user support mechanisms. However, two limitations complicate an integration of this data into applications:

- **First**, the data availability. The most important, yet most critical part of Transparency is the ContAct monitor. A monitoring unit requires extensive system resources and is very sensitive to modifications of the operating system which is being monitored. To really use such data in products, the manufacturers of the operating system should provide respective interfaces and security mechanisms for application developers who 1) build applications which deliver activity information 2) build applications which consume activity information.

- **Second**, data privacy, particularly in organizational settings. Activity data is a critical source of information as it is an overall performance and interest indicator, offering very sensible information about the subject. The ContAct monitor developed for the dissertation addresses these issues in three respects: 1) local system architecture, 2) data transparency and 3) configurability. Nevertheless, large scale solutions which cover more than one device require a server infrastructure and the central storage of activity data. Data security and mechanisms like privacy preserving machine learning and privacy preserving data mining [158] need to be considered in close alignment with legal aspects for the respective community [306]. Despite privacy issues which would need to be addressed, existing tendencies of life logging device usage indicate that people are willing to collect large amounts of data about their activities as long as they benefit from the data.

10.1.3 Characteristics of Systems to Support Memory

Although the relevance of prospective and retrospective memory support is out of the question, the actual effect of such solutions needs to be considered as well. The user group which worked with Transparency 2.0 completely trusted the tool. The participants did not question the data and they did not try to answer study questions based on their own memory once they had started working with the tool. These indicators suggest the willingness of users to accept such a tool as a specific type of transactive memory. Transactive memory refers to a type of memory which is distributed within a group. An individual does not memorize certain facts but rather only memorizes where to find it. The reliance on Google search queries when the participants recalled the work processes is an indicator for transactive memory resulting from internet usage [265].

A transactive memory for an individual’s past activities and knowledge actions which is seamlessly integrated into the daily work process not only helps information workers but makes them more dependent on the respective tool.

10.2 Outlook

Different aspects of the work provided in this thesis are promising directions for further investigation:

- **Knowledge actions:** The taxonomy of knowledge actions for information work at the computer workplace covers only a small facet of existing knowledge actions (cf. [221, 109]). Within this dissertation, knowledge actions are only used to group activities and describe work processes. A deeper investigation into knowledge actions, the most relevant desktop operations and support patterns for different knowledge actions are potential next steps.

- **Tension based system design methods:** The AT-SDM has been applied successfully within this dissertation. Other use cases show the capabilities of designing systems for collaborations based on the AT-SDM [76]. A deeper investigation into tension analysis patterns, especially with respect to individual and group ASMs would extend the usefulness of the design method.

- **Activity mining:** The discussed approaches for activity data show good results. Nevertheless, the approaches lack long-term strategies for data handling. Respective strategies need to be identified, evaluated and implemented. Additionally, a hierarchy of task clusters might be more appropriate for many information work scenarios, offering clusters of clusters to the users to categorize activities. Multinomial parameter estimation approaches as used in topic modeling could offer such functionalities.

- **Multiple devices:** Research that uses interaction histories from the different devices people interact with would be able to identify new classes of activities, the interplay of the different devices and striking media gaps.

A future direction of activity-based support to be highlighted explicitly is the integration of the community. Considering activity data from groups of users enables new types of support. Explicit recommendations for work processes can be generated based on best practices mined for different work processes. Group interactions can be simplified by providing additional background information (e.g., the circumstances information objects were created in).

Group activity data may positively influence the domain of business process management. Human tasks in business processes are a complex element within the process modeling. A task may be modeled as a workflow, thus limiting the individual's possibilities of enhancing the work processes based on personal experience. An alternative is modeling human tasks as a blind spot of the business process. Such an approach possibly complicates the execution for inexperienced persons. Activity data may provide support between these two extreme points. The data is a learning information storage, enhanced and updated based on the data set of each successful task execution. Therefore, research on activity-centric business process management might be an interesting next step [246][1].

[1] Related ideas have been sketched for agile business process management [244, 234].

A Background Data

A.1 Activity Data Collection

The relevance of activity data collection and processing techniques is the reason for a closer investigation of this topic. To address the problems of manual collection of user data, automated approaches have been considered early. Still, the mere collection of the data is not sufficient: “those attempting to represent user behavior must develop methods to abstract relevant information from a verbose interaction record, and must, therefore, identify what constitutes relevant information .”[90] To structure the information about the externalization collected by the state of the art review of support tools the different techniques are summarized based on the collection and the use of the data. For the externalization process, four different approaches are presented (see figure A.1). All externalizations create representations of activity, but they have different strength and require different effort.

- **Manual activity data creation:** The manual externalization is an obvious method to collect activity data. Eventually, the information worker or a consultant of a specific domain should have most information about activities which might occur. Nevertheless, the manual creation of data about activities is tedious and needs to consider that some activity information only emerges at the moment an activity is actually executed.

  The description of a task with little meta information like due date or priority is realized as a simple task management feature in many applications. Examples are Microsoft Outlook tasks, Lotus Notes Activities or getting things done tools like Remember The Milk [130]. More complex formalizations that include a notion of involved information objects, applications or the structure of work require much effort. Bailey specifies a system to describe user tasks in an XML format. Lesh applies the planning language UWL to describe goal schemes with related action schemes [155]. Cheikes proposes an activity notion of hierarchical expectation models to describe the execution of activities [54]. A comparable approach is the modelling of tasks based on grammars, in the sense that grammars can express hierarchical constructs. One example for grammar based task models is Activity Streams [178]. The disadvantage of complex procedures is the focus on recurring activities as other activities are not known beforehand. Task management tools are embedded in the work process and address many identified tensions, but even the maintenance of these tools is complex and triggers additional interruptions.

Sensor based approaches are an alternative way of identifying activities. Sensors are used to capture human activity. Sensors can be ubiquituous sensors like cameras, microphones, etc. Another class of sensors are software sensors that capture information about the human system interaction. One of the first applications of software sensors to create an interaction history of information work execution was described by Bannon [18] who tracked the command line input of users. For the domain of information work, the combination of software and ubiquituous sensors has been realized in the Kimura system [288, 289].

Based on software sensors different types of interaction histories can be created:

- **Programming by example for activity data creation:** Programming by example follows the idea of a macro recorder. The user records one activity and creates an interaction history which contains only elements of the one recorded activity. The recorded macro can be used to support tedious but repetitive work tasks based on automation [157, 64, 63]. The technique is only useful for repetitive activities which never differ.

- **Activity learning:** Activity learning uses annotated interaction histories to identify features that are similar for similar tasks. Therefore, it is a technique which is mainly used to train a system to identify tasks later on. As the technique trains on many instances of quasi similar activities, the trained system will use the most important features to identify tasks. The method requires an understanding of the tasks that occur and are intended to be identified. Comparison of techniques for identification: [102].

- **Activity mining:** Interaction histories that emerge during the daily interaction of a user with the computer are used to identify activities. The method is useful if only little information about the existing tasks is available. Approaches have been presented by [218, 33, 204].

Two important types of activity data usage can be distinguished. First, collected activity data is often visualized to help people to better understand their activities. Task management is a basic example. The recently used file list of most applications is an example of an automatically created visualization. A sparse interaction history that only focuses on file open operations of a specific application is automatically collected and visualized as a recently used list. The Microsoft Outlook Journal with a Gantt chart representation of used files is a more complex example of this visualization.

Second, collected activity data is used to identify the activity a user is working on. This is achieved by calculating a similarity between a representation of the newly collected activity data and a collection of formerly learned or mined activities. An important requirement for activity identification is the identification of activity switches. Without the identification of activity switches a data set that stands for different activities is used to identify a related task. The problem has been extensively discussed in the context of the TaskTracer system [255, 256, 257, 258]. Based on this identification automatic tasks can be triggered.

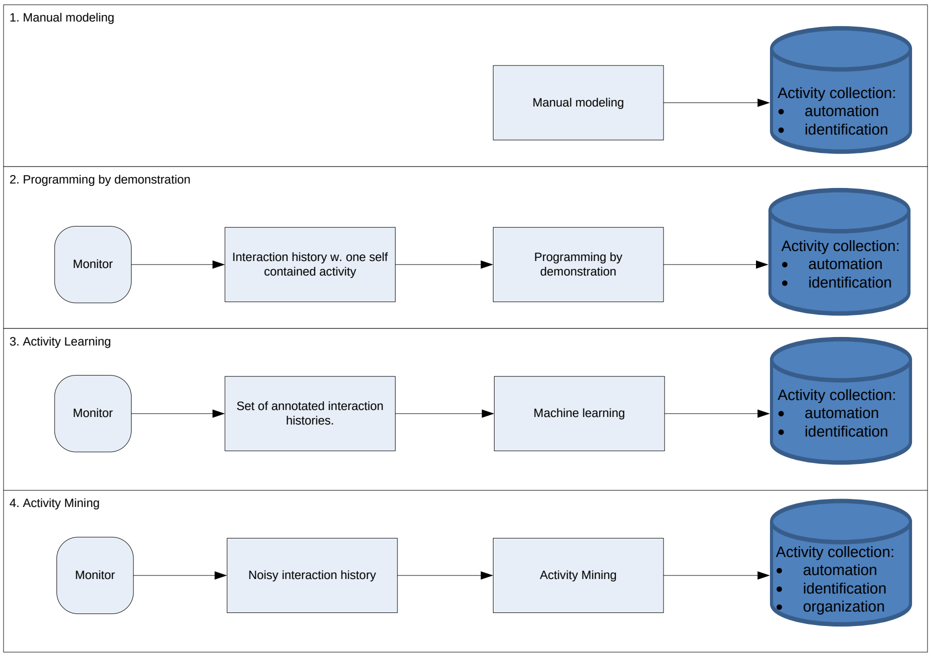

Figure A.1.: Means to collect activity data.

A.2 Knowledge Action Activity Systems

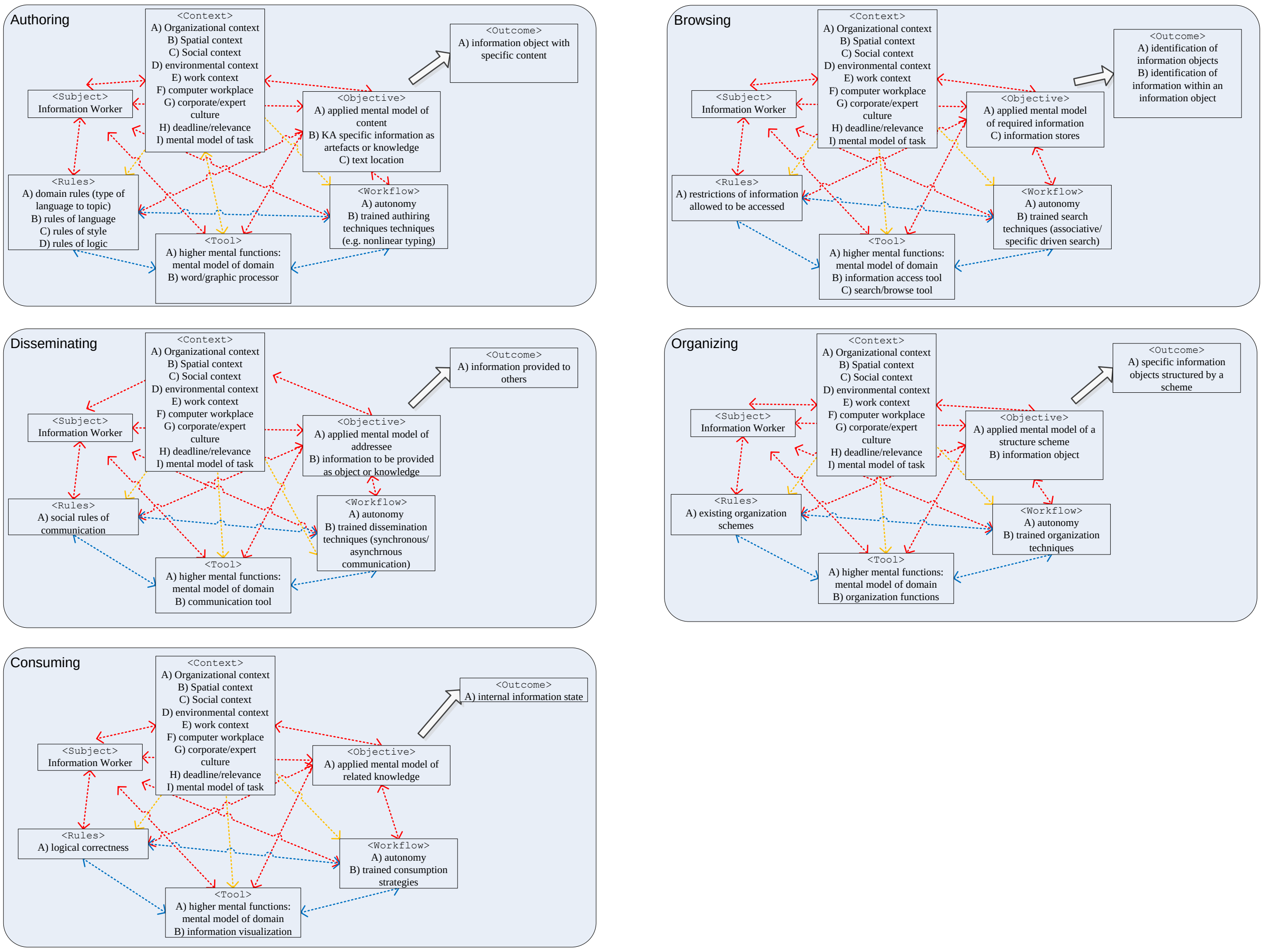

Figure A.2.: Knowledge action activity systems.

B Component Integration of Transparency 2.0

The following overview of the components shows the use of the ContAct monitor and an activity mining module within Transparency 2.0. The overall system architecture is composed of three different modules, comprising the ContAct monitor, an activity mining tool and the Transparency tool (see Figure B.1). The ContAct monitor contains components to realize the interaction history management process, including an interaction history monitor for data collection, an interaction data processor and an interaction data organizer. The generated activity data is structured by an instance of the computer work ontology (CWO) which is distributed to the activity mining component and Transparency 2.0. The activity mining component enhances the CWO ontology with task instance data and topic data, resulting from an activity mining and a topic extraction component. The final CWO ontology is consumed by Transparency which uses the data in a task management component, a recommender component and a history visualization component which stand for the three integrated support methods.

The components were realized using the .net framework and Java. The .net framework is used for monitoring the Windows operating system and visualizing the user interface, using the Windows Presentation Foundation [185]. Java is mainly used for ontology handling (using Jena [272]), complex event processing to identify desktop operations (using Drools Fusion [219]) and natural language processing (using UIMA with different plugins [271]). The .net and the Java part exchange data using CORBA [285] or via direct exchange of ontology files.

Transparency 2.0 is developed on top of the existing Tasks.show application [273]. Tasks.show showcases the WPF framework in a task management application. The existing structure provided a reasonable foundation for the development of Transparency 2.0. Images of the Transparency 2.0 application are given in the appendix, see Figure F.

Two processes coordinate the data exchange between the modules to provide Transparency permanently with updated activity data and regularly with mined activities:

- ***Permanent process*** ContAct permanently logs events and aggregates them to desktop operations and knowledge actions. Transparency subscribes this data stream and uses it for the interactive activity history and for the PASTREM recommender.

- ***Regular process*** The activity mining module regularly identifies activities in the identified knowledge actions and enhances the CWO ontology with this data. Updated CWO ontologies are provided as file to Transparency which uses this data for the task management module.

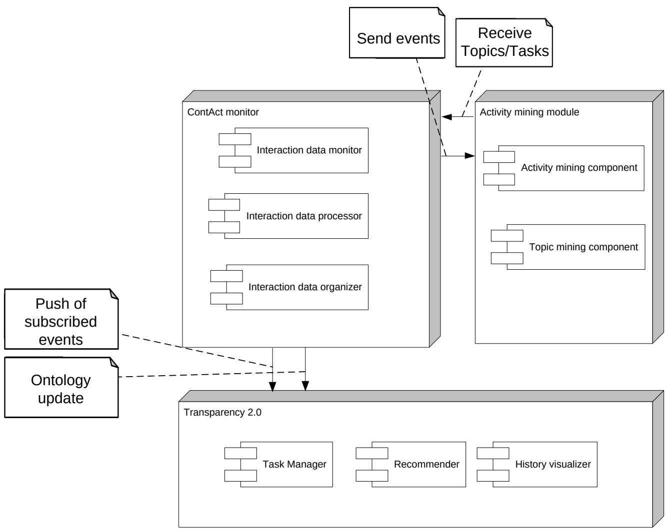

Figure B.1.: Component overview, including ContAct monitor, activity mining module and Transparency 2.0.

C Studies and Data Sets

Within the dissertation various evaluations are reported which are based on different data sets. This chapter gives and overview of all used data sets, the specific characteristics and their purpose. For each data set, the purpose is provided and the chapters within the dissertation which use the data set are listed.

- Explore work execution (uses data set 1)
- Derive knowledge actions (uses data set 1)
- Evaluate activity mining (uses data set 1 for calibration, data set 2 for gold standard analysis, data set 3 and 4 for real world evaluation)
- Transparency 1.0 (acceptance/work integration study)
- Transparency 2.0 (PASTREM recommender, recall study, acceptance/work integration study)

C.1 Data Set 1: Exploration Data Set – Controlled Mono Tasking Work Execution Data

Characteristics:

- Method:
  - Software sensor to create interaction history of work execution (used early version of ContAct monitor)
  - Video of work execution (Camtasia studio to capture computer interaction and user face)
  - Shadowing with note taking of work execution (Based on thinking aloud and observation peculiarities and milestones of work process)
  - Survey asking to renarrate the work process
- Type: Data collection in controlled work situation
- Participants: 20
- Captured execution time: 90 minutes task execution per participant
- Details: Participants executed a subset of seven predefined tasks. Task details are provided on paper and given in random order to the participants. After task finalization, the next task is given to a participant. Once a task is finalized, the participant fills out a survey sheet and specifies his work execution process. After 90 minutes the study ends, even if a participant was not able to execute all seven prepared tasks.

  Tasks were created based on focus group discussions to identify tasks the participants were familiar with.

Purpose:

- Used to identify knowledge actions and desktop operations (see section 3.3).
- Used to model heuristics to identify knowledge actions and desktop operations (see section 6.3).
- Used to calibrate algorithms (see section 7.3)

Task 1	Provide information on related work on individual topic
Task 2	Set up meeting to discuss conference paper review
Task 3	Decide on applicant invitation and communicate your decision
Task 4	Plan a trip and inform your colleague with all involved information
Task 5	Present a paper from a foreign language to your colleagues
Task 6	Find Application partners and experts for research project
Task 7	Search for Information on software functionality and save for later use

Table C.1.: Tasks used for the user study.

C.2 Data Set 2: Gold Standard Data Set – Controlled Multitasking Work Execution Data

Characteristics:

- Method:
  - Software sensor monitoring of work execution
  - Video of work execution
  - Shadowing with note taking of work execution
- Type: Data collection in controlled work situation
- Participants: 8
- Details: Participants execute five predefined tasks. Task details are provided on paper and given in random order to the participants. The study focuses on task switches. New tasks are given to the participants at random during the study, requiring the participants to switch to the new task and resuming the interrupted task once there is time left within the study. This study design results in a noisy interaction history to simulate real world multitasking and test activity mining algorithms.

  Tasks were created based on focus group discussions, to identify tasks the participants were familiar with.

Purpose:

- Create a gold standard for activity mining (see chapter 7). While data set 1 is used to calibrate thresholds for clustering algorithms, data set 2 is used to test against a gold standard.

C.3 Data Set 3: Activity Mining Data Set Small – Real World Work Execution Data

Characteristics:

- Method: Interaction histories resulting from information work, collected within 5 days
- Type: Data collection in open work situation
- Participants: 6

Purpose:

- Evaluate mined activities with respect to purity and recall effects (see chapter 7).

C.4 Data Set 4: Activity Mining Data Set Large – Real Work Execution Data Collected Over Long Period of Time

Characteristics:

- Method: Logs of information work execution data, between 14 days and 2 months of data
- Type: Data collection in open work situation
- Participants: 2

Purpose:

- Evaluation of the PASTREM recommender (see chapter 9.3).

C.5 Data Set 5: Transparency 2.0 Recall Data Set – Controlled Multitasking Work Execution Data

Characteristics:

- Method:
  - Software sensor monitoring of work execution
  - Video of work execution
  - Shadowing with note taking of work execution
- Type: Data collection in controlled work situation
- Participants: 8
- Details: Participants execute five predefined tasks. Task details are provided on paper and given in random order to the participants. The study focuses on task switches. New tasks are given to the participants at random during the study, requiring the participants to switch to the new task and resume the interrupted task once there is time left within the study. The study organizer ends the study after approx. 40 minutes of work and ensures that each participant has at least 2 unfinished tasks.

  Tasks were created based on focus group discussions to identify tasks the participants were familiar with.

Purpose:

- Evaluate prospective and retrospective memory of work execution captured in the data set in a controlled study with an experimental group (Transparency 2.0 users) and a control/comparison group (non-Transparency 2.0 users). See chapter 9.5.

C.6 Data Set 6: Transparency 1.0 Test Data Set – Integration of Transparency 1.0 into the daily work processes

Characteristics:

- Method:
  - Software sensor monitoring of work execution
  - Interviews
- Type: Data collection in controlled work situation
- Participants: 9
- Details: For the evaluation of Transparency 1.0, nine users were recruited using convenience sampling. 7 were male, 2 female, with ages between 26 and 38. Users were either researchers or managers at an IT vendor and had significant IT experience. Their work included a high degree of self organization, involvement in multiple projects and commitment to an expert culture, thus fitting the profile of the information worker very well. None of them had used Transparency before. Users tested Transparency 1.0 for two weeks during their daily work activities (i.e., for 10 work days).

  At the beginning of the study, they received a demonstration of Transparency's features and were asked to fill out one questionnaire regarding their personal working style and one regarding their impression of Transparency. They were asked to complete the latter again after the study had been completed. Additionally, an unstructured interview was conducted after the study. The interviews were evaluated based on clustering statements.

  Transparency 1.0 contains a visualization of the user work process, an activity data management feature and a recommender.

Purpose:

- Investigate the user acceptance of the Transparency 1.0 tool (see chapter 9.4).

D Measures

In the following table D.1, all possible cases for the classification of an item are displayed. The terms true positive, true negative, false positive and false negative are used to compare the identified class of an item to the actually correct class.

	correct class 1	correct class 2
identified class 1	**true positive**	**false positive**
identified class 2	**false negative**	**false positive**

Table D.1.: Classification result options.

Precision:
$precision = \frac{NumberOfTruePositives}{NumberOfTruePositives+NumberOfFalsePositives}$
For clustering: The fraction of items in a cluster which belong to the activity label assigned to the corresponding cluster.
Recall:
$recall = \frac{NumberOfTruePositives}{NumberOfTruePositives+NumberOfFalseNegatives}$
For clustering: The fraction of items that belong to an activity label and appear in the corresponding cluster

E Transparency 1.0

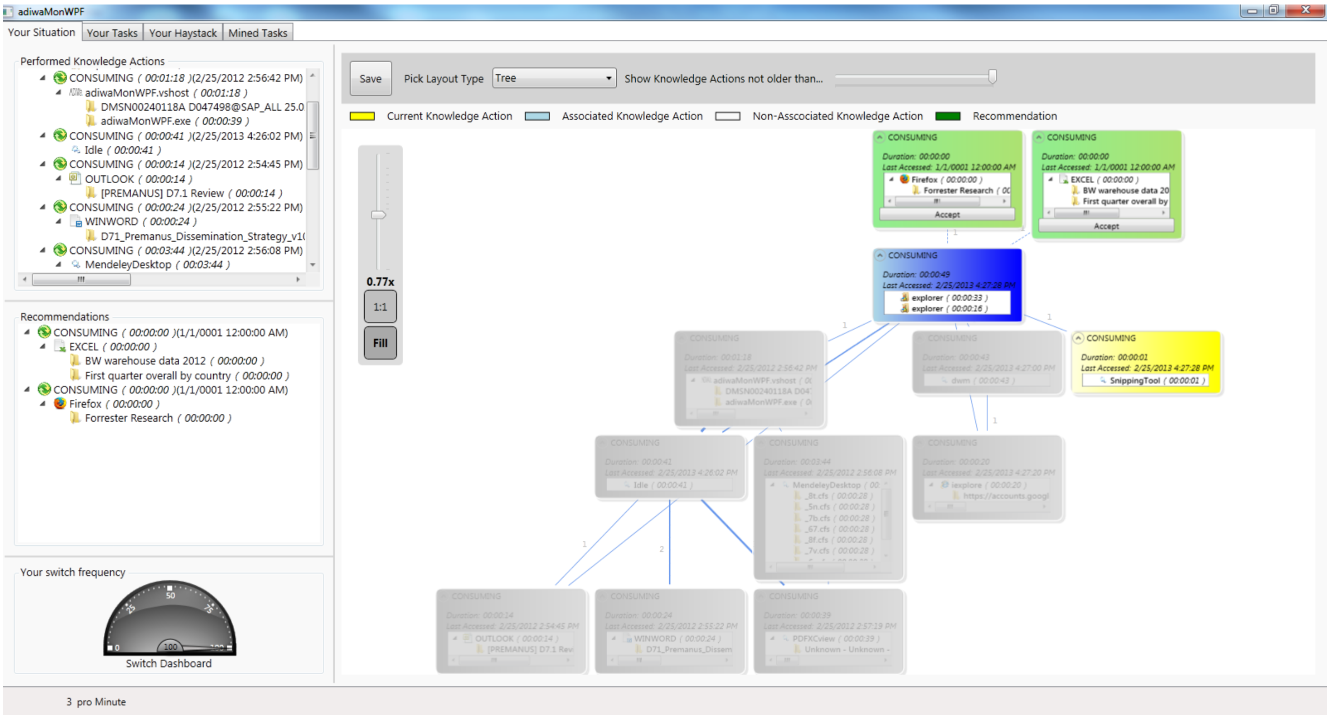

Figure E.1.: Situation overview. A graph with all knowledge actions performed by the user and additional recommendations. The user can filter the graph based on the access duration and the date of last recent access.

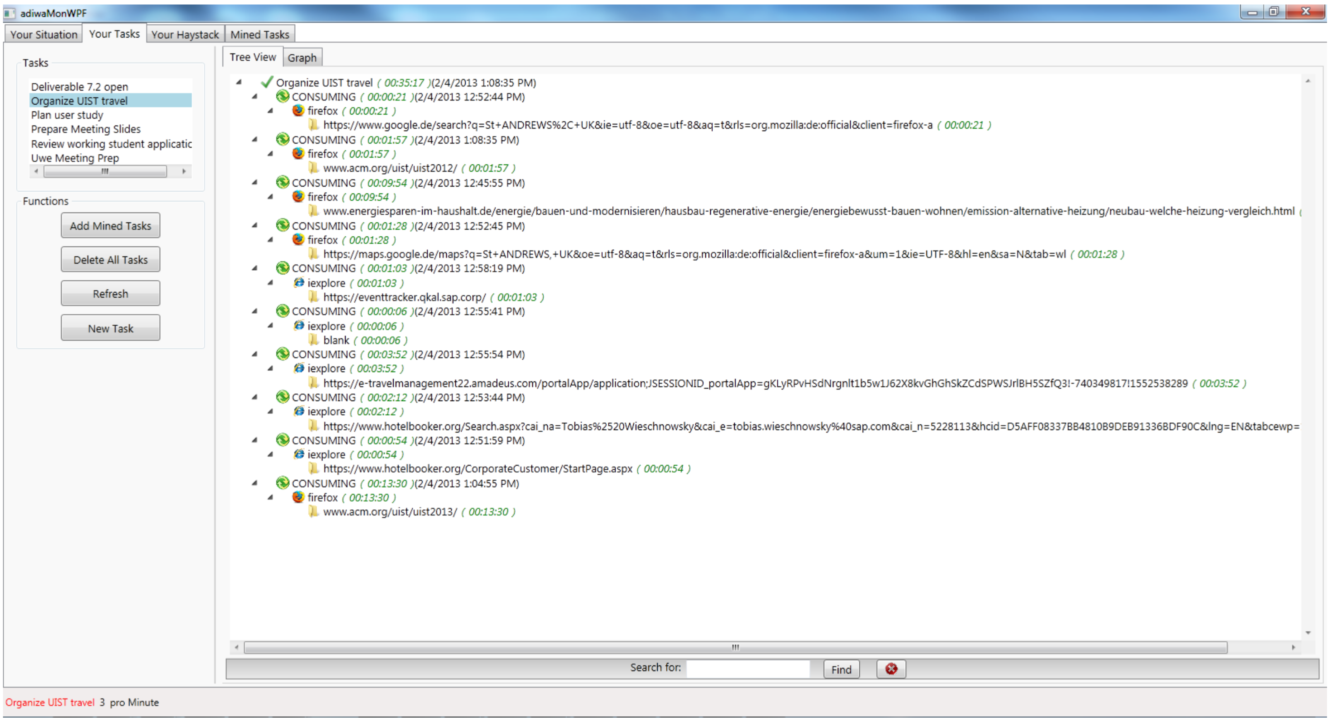

Figure E.2.: A user maintained list of executed activities with information about the duration and access to a respective graph visualization of the activity.

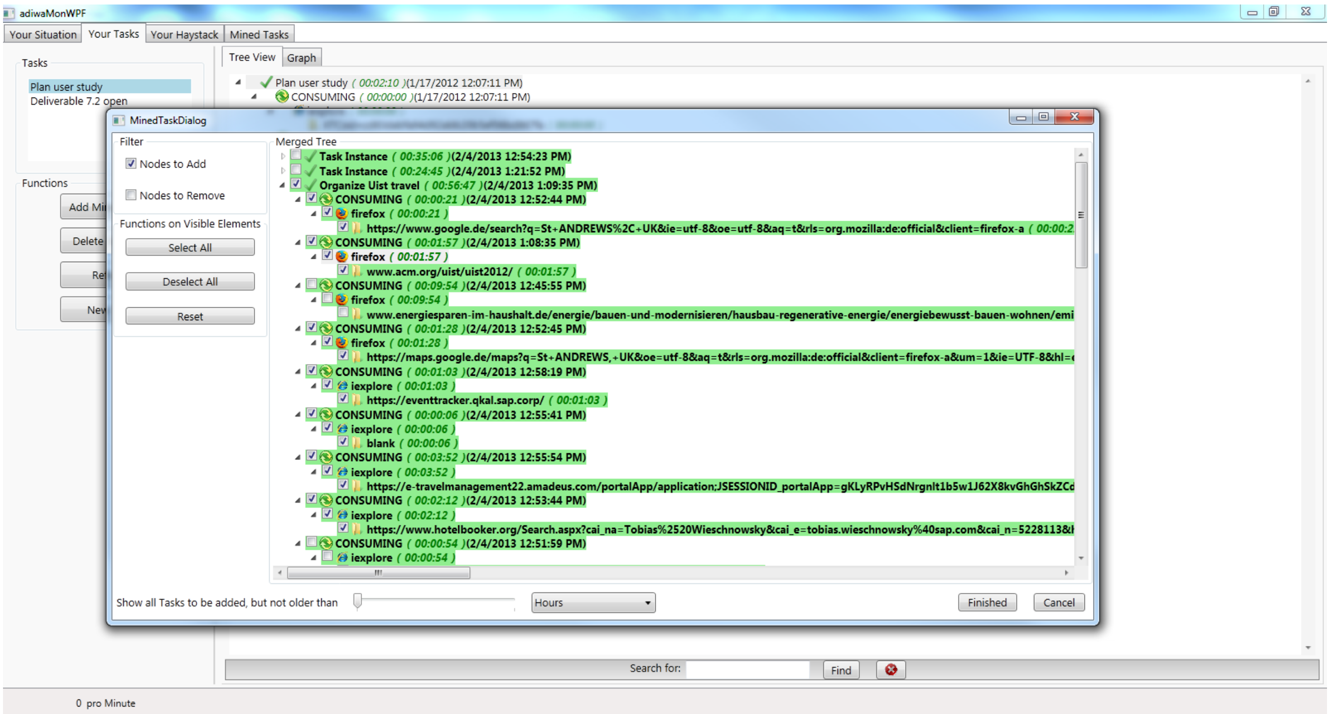

Figure E.3.: A dialog to add activities to the maintained activities dialog. The system presents all mined activities. Merge operations are recommended. This is indicated by the green boxes added to existing activities.

F Transparency 2.0

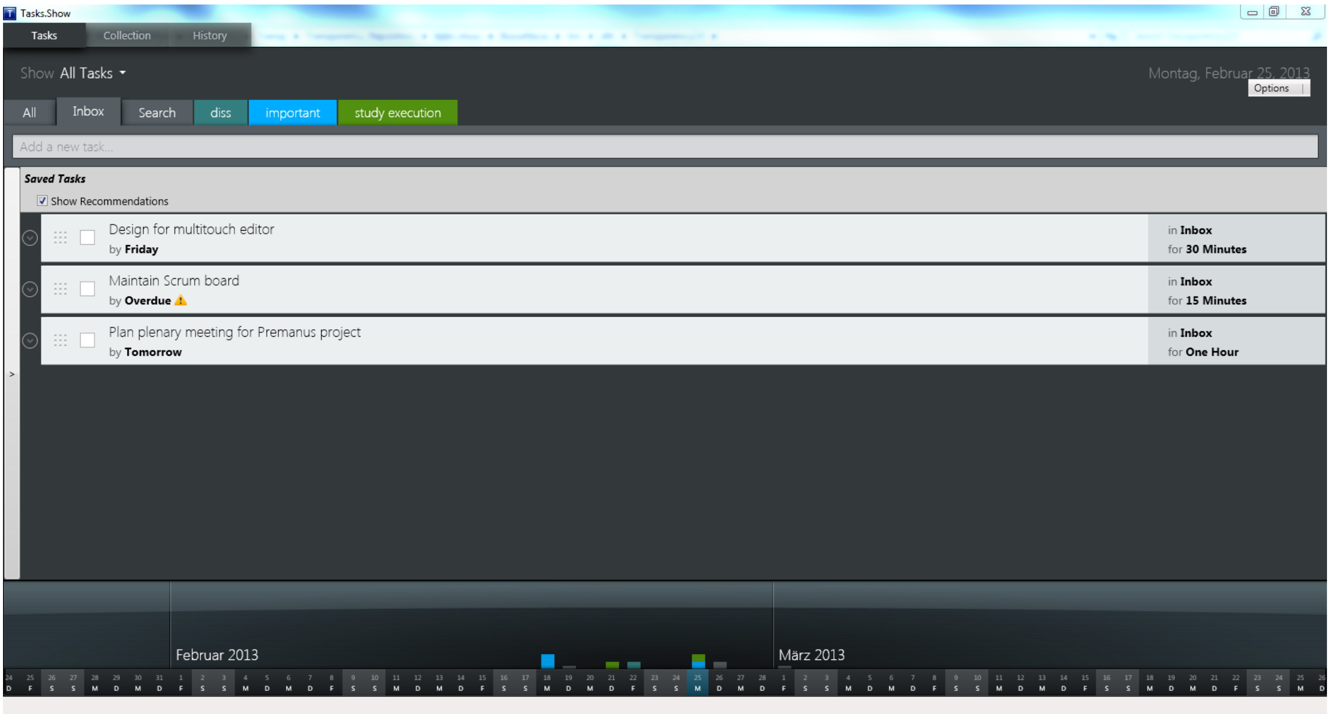

Figure F.1.: A task list of the user in Transparency 2.0. The original view was developed for [273].

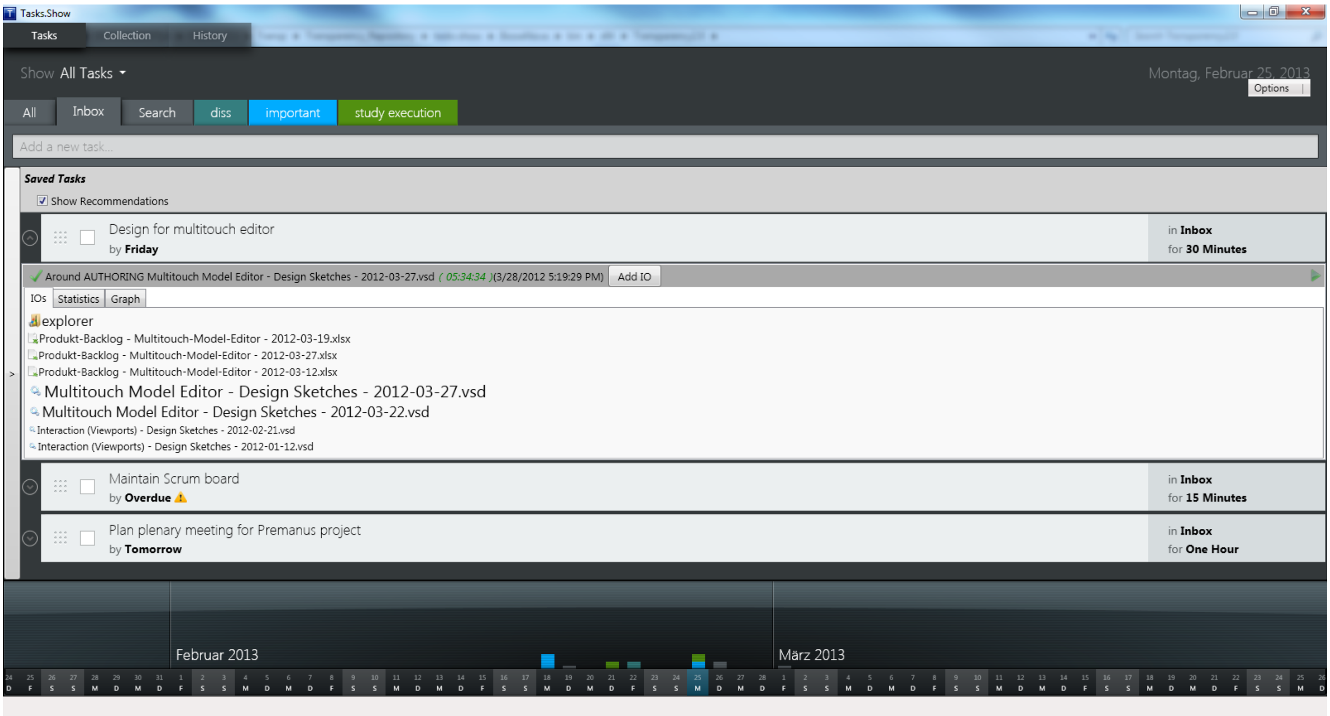

Figure F.2.: A task object is uncollapsed to access activity data. From the three tabs *Information objects*, *Statistics* and *Graph*, the *Information object* tab is selected. An overview of all information objects associated with the activity is provided and the activity time per information object is visualized based on the size of the object. The user can manually add new information objects which are automatically tracked for the respective activity. If the system identifies information object usages which share similarities with the respective activity, the system recommends to add them to the activity.

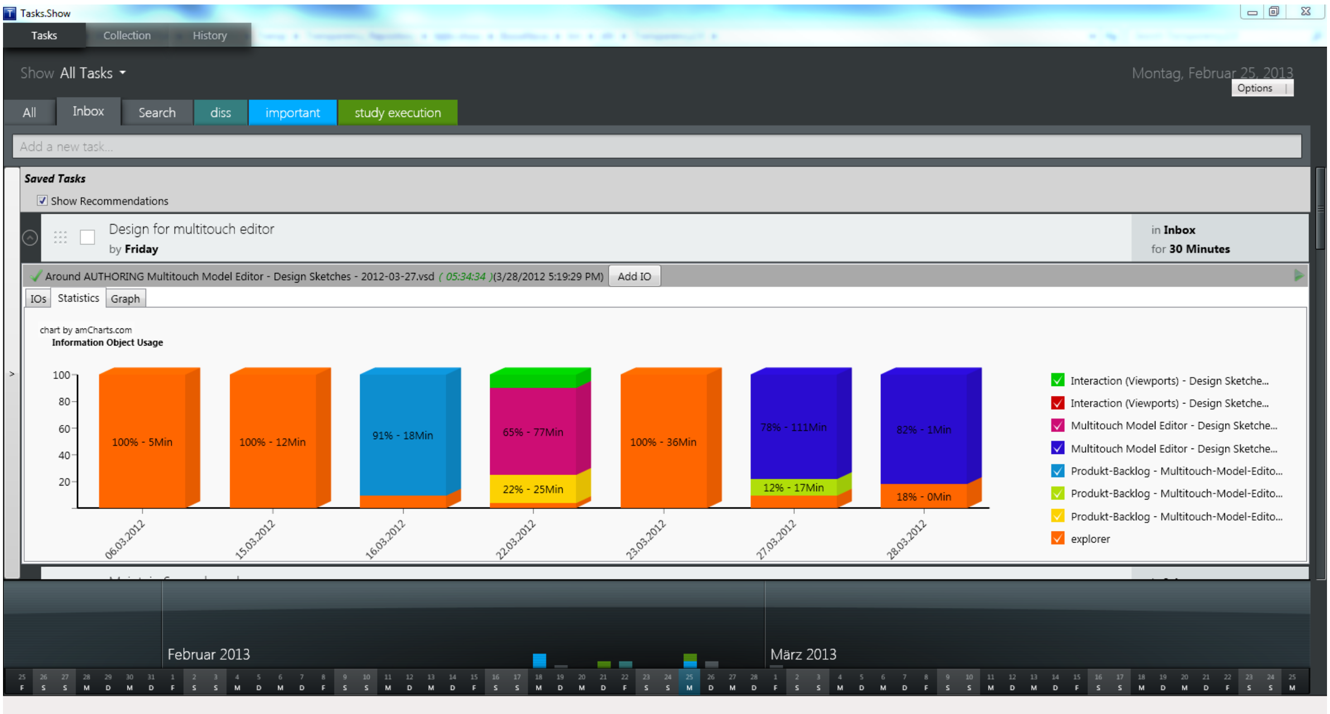

Figure F.3.: A task object is uncollapsed to access activity data. From the three tabs *Information objects*, *Statistics* and *Graph*, the *Statistics* tab is selected. An overview of the different timeframes during which the user worked on information objects is provided.

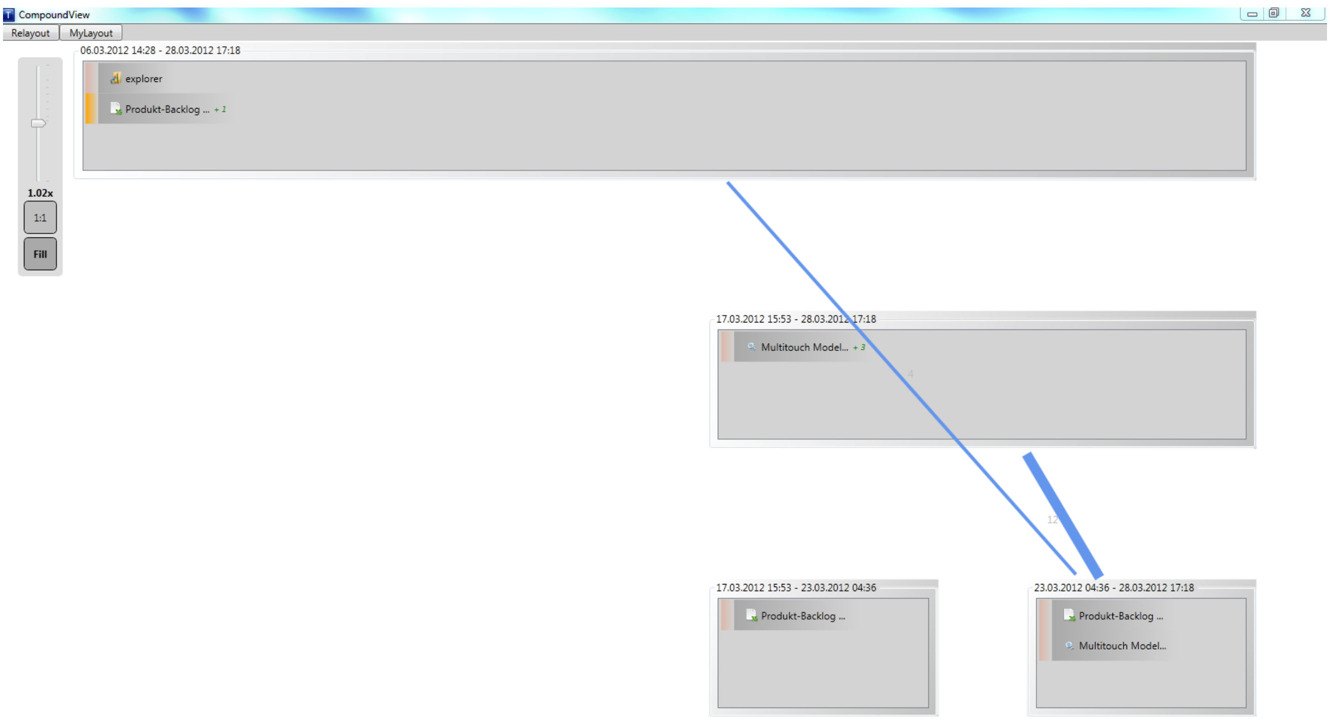

Figure F.4.: The compound graph provides a fuzzy process knowledge of the work process associated with the activity. The upper box holds those information objects which were used during the whole tracked work time. The box at the lower right only holds information which was used at the end of the tracked work time.

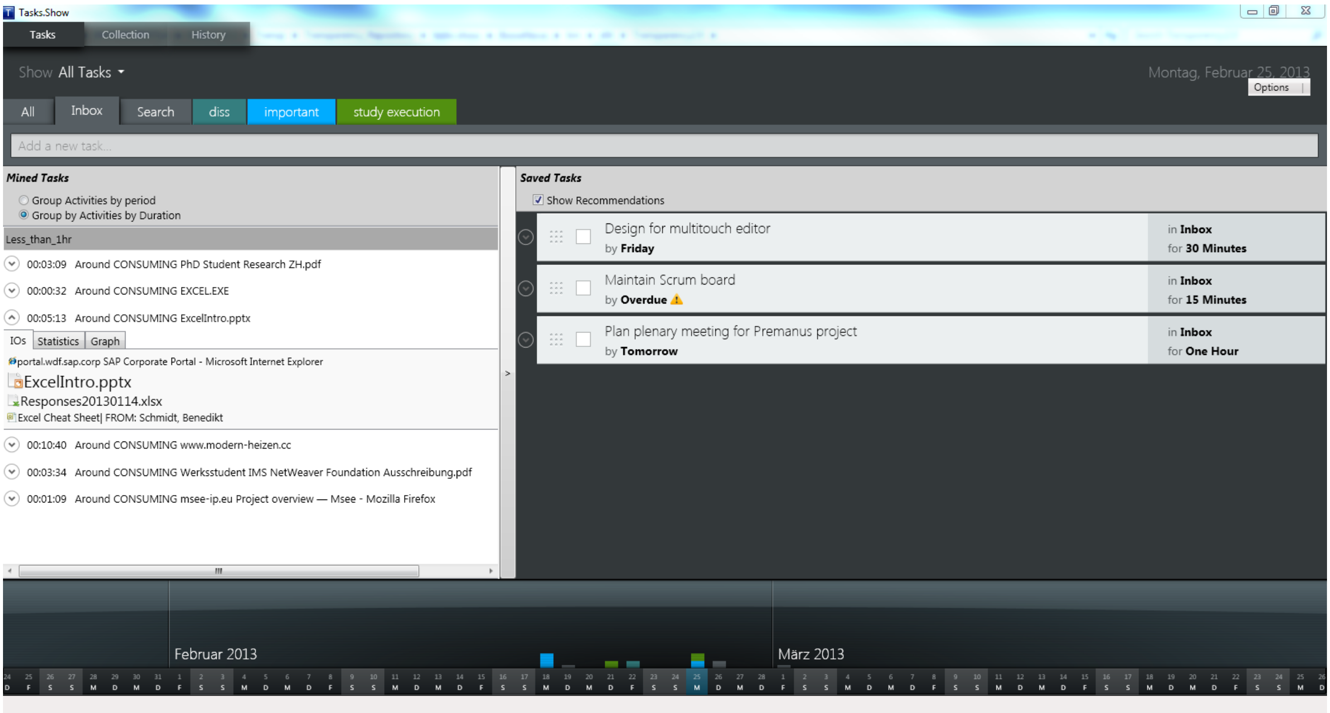

Figure F.5.: The mined activities are displayed next to the existing tasks. The user can drag and drop activity data sets and single activity elements into the tasks. The mined activities can be sorted and searched based on criteria like overall duration, last work on the activities and information object name strings.

Figure F.6.: The activity history as a dynamic graph organized by a timeline. The user can select time segments, can filter the displayed knowledge actions based on their overall duration and their age. Each node contains keywords which unfold when the user hovers over the node. Keywords and direct text entry can trigger search within the timeline to identify time segments relevant for the search term. Single nodes can be selected and only their neighbors are displayed to simplify exploration of large graphs. When the time segment is shifted the modification of the work process is animated as some nodes remain in the display, other nodes appear or disappear.

G Process

G.1 Interaction History


```xml
<event eventName="OUTLOOK_MAIL_CHANGED" atTime="12.12.2012 10:40:21.284" eventCategory="Application">
    <eventattributes>
    <eventattribute name="from" type="String" value="Quaye, George" />
    <eventattribute name="to" type="String" value="Schmidt, Benedikt " />
    <eventattribute name="subject" type="String" value="RE: Campus Connect Demo: Transparency" />
    <eventattribute name="content" type="String" value="Hello Benedikt, Sure… can we do it very early for me, 6:30am PST Thanks… George From: Schmidt, Benedikt Sent: Tuesday,
    <eventattribute name="entryID" type="String" value="00000000B1F185E517907240BB4EEFDDCDA260520700E88BF24C0A1C994A8A256F02CD9DAF69000000930D6C0000E53DD7C207334C44A42F4B934
    <eventattribute name="storeID" type="String" value="0000000038A1BB1005E5101AA1BB08002B2A56C20000454D534D44422E444C4C00000000000000001B55FA20AA6611CD9BC800AA002FC45A0C000
    </eventattributes>
</event>

<event eventName="OUTLOOK_MAIL_CHANGED" atTime="12.12.2012 10:40:32.459" eventCategory="Application">
    <eventattributes>
    <eventattribute name="from" type="String" value="benedikt.schmidt@sap.com" />
    <eventattribute name="to" type="String" value="Goeb, Andreas " />
    <eventattribute name="subject" type="String" value="Emailing: UISpy.zip" />
    <eventattribute name="content" type="String" value="  FYIYour message is ready to be sent with the following file or link attachments:UISpy.zipNote: To protect against c
    <eventattribute name="entryID" type="String" value="00000000B1F185E517907240BB4EEFDDCDA260520700E88BF24C0A1C994A8A256F02CD9DAF69000000930D6E0000E789FBB74A472B4EBEFC465C7
    <eventattribute name="storeID" type="String" value="0000000038A1BB1005E5101AA1BB08002B2A56C20000454D534D44422E444C4C00000000000000001B55FA20AA6611CD9BC800AA002FC45A0C000
    </eventattributes>
</event>

<event eventName="FOREGROUND_WINDOW_CHANGED" atTime="12.12.2012 10:40:34.398" eventCategory="Process">
    <eventattributes>
    <eventattribute name="processname" type="String" value="iexplore" />
    <eventattribute name="windowtitle" type="String" value="" />
    <eventattribute name="processurl" type="String" value="C:\Program Files (x86)\Internet Explorer\iexplore.exe" />
    <eventattribute name="content" type="String" value="   /* &lt;![CDATA[ */var targetRegion='de-0';homepageBegin();/* ]]&gt; */                                K
    <eventattribute name="associatedFile" type="String" value="::::http://www.gmx.net/" />
    </eventattributes>
</event>

<event eventName="FOREGROUND_WINDOW_CHANGED" atTime="12.12.2012 10:41:29.576" eventCategory="Process"><eventattributes>
    <eventattribute name="processname" type="String" value="iexplore" />
    <eventattribute name="windowtitle" type="String" value="PST: Pacific Time (USA &amp; Canada). What's the current time now in PST? / PDT in Summer - Microsoft Internet Ex
    <eventattribute name="processurl" type="String" value="C:\Program Files (x86)\Internet Explorer\iexplore.exe" />
    <eventattribute name="content" type="String" value="      Pacific Time         var addthis_config = {&quot;data_track_clickback&quot;:true};                GreenwichMe
<eventattribute name="associatedFile" type="String" value="::::http://wwp.greenwichmeantime.com/time-zone/usa/pacific-time/" /></eventattributes></event>

<event eventName="FOREGROUND_WINDOW_CHANGED" atTime="12.12.2012 10:41:35.512" eventCategory="Process"><eventattributes>
    <eventattribute name="processname" type="String" value="AcroRd32" />
    <eventattribute name="windowtitle" type="String" value="Internship SAP Research GEYSERS.pdf - Adobe Reader" />
    <eventattribute name="associatedFile" type="String" value="C:\Documents and Settings\evaluationuser4\2010_JKM_KWRS_v0.9.pdf" />
    <eventattribute name="content" type="String" value="   Knowledge worker roles and activities &#xD;&#xD;Anonymous&#xB;&#xD;&#xD;Abstract - This paper proposes a typology
    </eventattributes>
</event>
```


Figure G.1.: The mined activities are displayed next to the existing tasks. The user can drag and drop activity data sets and single activity elements into the tasks. The mined activities can be sorted and searched based on criteria like overall duration, the last work on the activities and information object name strings.

G.2 Rules

G.3 Basic Classes

```
rule "FocusWindow ByTaskBar" salience 15

    when
        $e1 : Event( eventName == "FocusedAutomationElementByMouse")
        Attribute ( name == "name", $name : value) from $e1.attributes
        Attribute ( name == "firstnamedparent", $pa : value) from $e1.attributes
        Attribute ( name == "localizedControlType", $ct : value) from $e1.attributes
        eval ($pa == "Running Applications" && $ct == "button")

        $e2 : Event ( this after [-1s, 3s] $e1, eventName == "FOREGROUND_WINDOW_CHANGED")
        Attribute( name == "windowtitle", $windowtitle : value) from $e2.attributes
        Attribute( name == "processname", $processname : value) from $e2.attributes
        //Attribute( name == "processurl", $processurl : value) from $e2.attributes
        Attribute( name == "associatedFile", $associatedFile : value) from $e2.attributes
        eval ($windowtitle == $name && !$windowtitle.contains(":\\"))
    then
        /*DesktopAction dA = new DesktopAction($e1.timeStamp, $e2.timeStamp, DAAction.FOCUSING, DAObject.WINDOW );
        dA.putAttribute("process", $processname);
        dA.putAttribute("windowtitle", $windowtitle);
        dA.putAttribute("associatedFile", $associatedFile);
        //dA.putAttribute("processurl", $processurl);
        dA.putAttribute("how", "by taskbar");
        dA.addEvent($e1);
        dA.addEvent($e2);
        dActions.add(dA);*/
end

rule "FocusWindow" salience 10

    when
        $e1 : Event ( eventName == "FOREGROUND_WINDOW_CHANGED")
        Attribute( name == "windowtitle", $windowtitle : value) from $e1.attributes
        Attribute( name == "processname", $processname : value) from $e1.attributes
        Attribute( name == "associatedFile", $associatedFile : value) from $e1.attributes
        //Attribute( name == "processurl", $processurl : value) from $e1.attributes
        eval(!$windowtitle.contains(":\\"))
        eval(!($windowtitle.equals("") && $processname.equals("")))
        eval(!($windowtitle.equals("") && $processname.equals("explorer")))
    then
        DesktopAction dA = new DesktopAction($e1.timeStamp, $e1.timeStamp, DAAction.FOCUSING, DAObject.WINDOW );
        dA.putAttribute("process", $processname);
        dA.putAttribute("windowtitle", $windowtitle);
        dA.putAttribute("associatedFile", $associatedFile);
        //dA.putAttribute("processurl", $processurl);
        dA.addEvent($e1);
        dActions.add(dA);
end
```

Figure G.2.: Two example rules which identify desktop operations based on a foreground window change. Drools notion is used.

classes.PNG

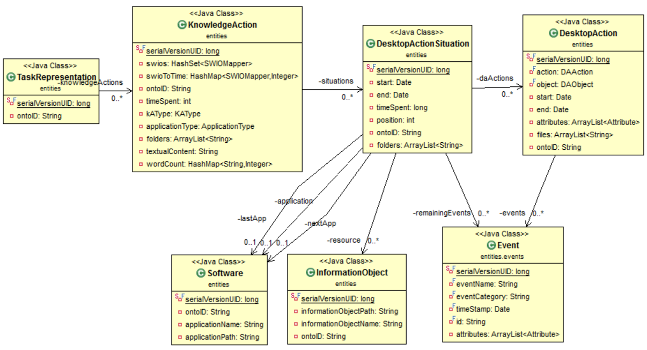

Figure G.3.: Basic classes for the activity structure: an activity representation, the desktop operation situation, the desktop operation and the event.

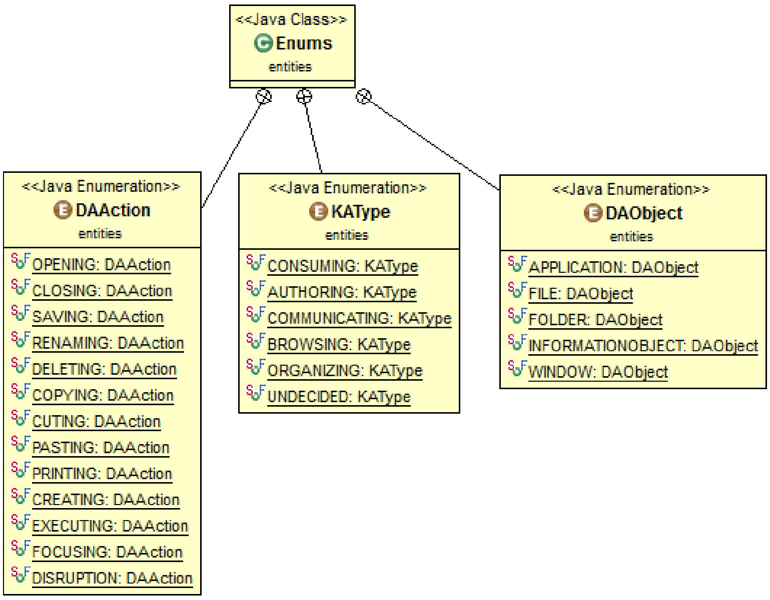

Figure G.4.: Enumerations which are used to classify knowledge actions and desktop operations.

H Curriculum vitae[2]

• 2002 – 2008	• Diplom Medienwissenschaftler (Media Studies Degree), Universität Paderborn
• 2003 – 2007	• Bachelor Of Computer Science, Universität Paderborn
• 2007 – 2009	• Diplom Informatiker (Computer Science Degree), Universität Paderborn
• 2008 – 2009	• Thesis Student, SAP Research Karlsruhe, SAP AG
• 2009 – 2013	• Research Associate, SAP Research Darmstadt, SAP AG
• 2009 – 2013	• PhD Student, Technische Universität Darmstadt

[2] Gemäß §20 Abs. 3 der Promotionsordnung der TU Darmstadt

Glossary

ART action regulation theory. 7, 11, 16–18, 22, 25, 27, 29, 31, 49, 55, 57–59, 61, 71

ASM activity system model. xix, 57–71, 73–75, 77–79, 81–85, 87, 91, 96, 98, 99, 177, 180, 181

AT activity-theory. 7, 11–16, 18, 22, 23, 27, 29, 45, 46, 49, 55, 57–61, 74, 177

AT-SDM activity theory based system design method. 7, 8, 55, 58, 59, 61, 68–71, 74, 79, 84, 91, 98, 99, 149, 177, 180, 181

UCD user-centered design. 6–8, 55–57, 70, 71, 98, 99, 140, 149, 165, 166, 175, 177–179

Bibliography

[1] P. D. Adamczyk and B. P. Bailey. If Not Now , When ?: The Effects of Interruption at Different Moments Within Task Execution. 6(1):271–278, 2004.

[2] E. Adar, D. Karger, and L. Stein. Haystack: Per-user information environments. In *Proceedings of the eighth international conference on Information and knowledge management*, pages 413–422. ACM Press, 1999.

[3] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 33(6):81–749, June 2005.

[4] G. Adomavicius and A. Tuzhilin. Context-Aware Recommender Systems. In *Context*, pages 1–37. Springer, 2008.

[5] W. Agresti. The conventional software life-cycle model: its evolution and assumptions. In *New Paradigms for Software Development*, pages 2–5. IEEE Computer Society, 1986.

[6] D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, January 1991.

[7] J. Anderson. *Rules of the mind*. Lawrence Erlbaum, 1993.

[8] J. Annett and K. Duncan. Task analysis and training design. 1967.

[9] S. Aral and E. Brynjolfsson. Information, technology and information worker productivity: Task level evidence. 2007.

[10] J. Atkinson. The achievement motive and recall of interrupted and completed tasks. *Journal of Experimental Psychology*, 46(6):381–390, 1953.

[11] J. Austin. *How to do things with words*. Oxford University Press, Oxford, 1975.

[12] D. Avison and G. Fitzgerald. *Information Systems Development: Methodologies, Techniques & Tools*. McGraw-Hill, 2006.

[13] A. Baddeley. Depth of processing, context, and face recognition. *Canadian Journal of psychology*, 36(2):148–164, 1982.

[14] A. Baddeley and N. Thomson. Word length and the structure of short-term memory. *Journal of verbal learning and*, 6(14):575–589, 1975.

[15] A. Baddeley. Working memory: looking back and looking forward. *Nature reviews. Neuroscience*, 4(10):829–39, October 2003.

[16] B. Bailey, P. Adamczyk, T. Chang, and N. Chilson. A framework for specifying and monitoring user tasks. *Computers in Human Behavior*, 22(4):709–732, 2006.

[17] B. P. Bailey and S. T. Iqbal. Understanding changes in mental workload during execution of goal-directed tasks and its application for interruption management. *ACM Transactions on Computer-Human Interaction*, 14(4):1–28, January 2008.

[18] L. Bannon, A. Cypher, S. Greenspan, and M. Monty. Evaluation and analysis of users' activity organization. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, number December, pages 54–57. ACM Press, 1983.

[19] J. Bao, C. Lyon, P. Lane, W. Ji, and J. Malcolm. Comparing Different Text Similarity Methods. *UH Computer Science Technical Report 461*, 2007.

[20] J. Bardram, J. Bunde-Pedersen, and M. Soegaard. Support for activity-based computing in a personal computing operating system. In *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06*, page 211, New York, New York, USA, 2006. ACM Press.

[21] J. Bardram and A. Doryab. Activity Analysis - Applying Activity Theory to Analyze Complex Work in Hospitals. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, pages 455–464. ACM Press, 2011.

[22] S. Barth. Self-Organization: Taking a Personal Approach to KM. In *Knowledge Management Tools and Techniques: Practitioners and Experts Evaluate KM Solutions*, pages 347–364. Butterworth-Heinemann, Boston, 2004.

[23] V. Bellotti. Managing Activities with TV-ACTA : TaskVista and Activity- Centered Task Assistant. In *Personal Information Management Workshop*, pages 8–11. ACM Press, 2006.

[24] V. Bellotti, B. Dalal, N. Good, P. Flynn, D. Bobrow, and N. Ducheneaut. What a to-do: studies of task management towards the design of a personal task list manager. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, volume 6, pages 735–742. ACM Press, 2004.

[25] J. R. Beniger. *The control revolution - Technological and Economic Origins of the Information Society*. Harvard University Press, Cambridge, 1986.

[26] M. Bernstein. Knowledge Work 2020, 2010. http://www.slideshare.net/PARCInc/knowledge-work-2020.

[27] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(4-5):993–1022, May 2003.

[28] S. Bø dker. *Through the Interface: A Human Activity Approach to User Interface Design*. Lawrence Erlbaum Assoc Inc, 1991.

[29] B. Boehm. A spiral model of software development and enhancement. *Computer*, 21(5):61–72, 1988.

[30] J. Boyce. Track your time in Outlook using Journal. http://office.microsoft.com/en-us/outlook-help/track-your-time-in-outlook-using-journal-HA010034346.aspx.

[31] R. Braune and D. Christopher. Time-sharing revisited: Test of a componential model for the assessment of individual differences. *Ergonomics*, 29(11):1399—-1414, 1986.

[32] O. Brdiczka. From documents to tasks: deriving user tasks from document usage patterns. In *Proceedings of the 15th international conference on Intelligent user interfaces*, pages 3–6. ACM Press, 2010.

[33] O. Brdiczka, N. Su, and B. Begole. Using temporal patterns (t-patterns) to derive stress factors of routine tasks. In *CHI 2009*, pages 4081–4086. ACM Press, 2009.

[34] O. Brdiczka, N. Su, and J. Begole. Temporal task footprinting: identifying routine tasks by their temporal patterns. In *Proceedings of the 15th international conference on Intelligent user interfaces*. ACM Press, 2010.

[35] J. Brixey, D. Robinson, C. Johnson, T. Johnson, J. Turley, and J. Zhang. A concept analysis of the phenomenon interruption. *Advances in Nursing Science*, 30(1):E26–E42, 2006.

[36] J. J. Brixey, Z. Tang, D. J. Robinson, C. W. Johnson, T. R. Johnson, J. P. Turley, V. L. Patel, and J. Zhang. Interruptions in a level one trauma center: a case study. *International journal of medical informatics*, 77(4):235–41, May 2008.

[37] A. B. Brush, B. R. Meyers, D. S. Tan, and M. Czerwinski. Understanding memory triggers for task tracking. *Proceedings of the SIGCHI conference on Human factors in computing systems*, page 947, 2007.

[38] P. Brusilovsky and D. W. Cooper. Domain, task, and user models for an adaptive hypermedia performance support system. In *Proceedings of the 7th international conference on Intelligent user interfaces - IUI '02*, page 23, New York, New York, USA, 2002. ACM Press.

[39] K. Burke. Language as symbolic action: Essays on life, literature, and method, 1968.

[40] I. Burmistrov. Interruptions in human-computer interactions. http://interruptions.net/.

[41] K. Byström. Task complexity affects information seeking and use. *Information Processing & Management*, 31(2):191–213, April 1995.

[42] K. Byström. *TASK COMPLEXITY , INFORMATION TYPES AND INFORMATION SOURCES : Examination of Relationships*. PhD thesis, University of Tampere, 1999.

[43] K. Byström and P. Hansen. Conceptual framework for tasks in information studies. *Journal of the American Society for Information Science and Technology*, 56(10):1050–1061, 2005.

[44] J. Canny. GaP: a factor model for discrete data. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM Press, 2004.

[45] S. K. Card, A. Newell, and T. P. Moran. *The Psychology of Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1983.

[46] D. Carmel, H. Roitman, and N. Zwerdling. Enhancing cluster labeling using wikipedia. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, page 139, New York, New York, USA, 2009. ACM Press.

[47] M. Castells. *The Rise of the Network Society, The Information Age: Economy, Society and Culture Vol. I.* Blackwell, Cambridge, MA; Oxford, UK, 1996.

[48] T. Catarci, A. Dix, A. Katifori, G. Lepouras, and A. Poggi. Task-centered Information Management. In *Digital Libraries: Research and Development*, pages 197–206. Springer, 2007.

[49] T. Catarci, R. La, B. Habegger, and A. Poggi. Intelligent user task oriented systems. In *Proceedings of the Second SIGIR Workshop on Personal Information Management*, pages 20–23. Springer, 2006.

[50] J. Cellier and H. Eyrolle. Interference between switched tasks. *Ergonomics*, 35(1):26–36, 1992.

[51] B. Chandrasekaran, J. Josephson, and V. Benjamins. The ontology of tasks and methods. *AAAI Technical Report SS-97-06.*, pages 1–25, 1997.

[52] K. M. Chandy. Event-Driven Applications : Costs , Benefits and Design Approaches. In *Gartner Application Integration and Web Services Summit*, 2006.

[53] D. Chau, B. Myers, and A. Faulring. What to do when search fails: finding information by association. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2008.

[54] B. Cheikes, M. Geier, R. Hyland, F. Linton, and L. Rodi. Embedded training for complex information systems. *International Journal of Artificial Intelligence in Education*, 10:314–334, 1999.

[55] P.-S. Cheung, R. Huang, and W. Lam. Financial activity mining from online multilingual news. In *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004.*, pages 267–271. IEEE, 2004.

[56] A. Cheyer, J. Park, and R. Giuli. IRIS: Integrate, Relate. Infer. Share. 2005.

[57] S. Cohen. After effects of stress on human performance and social behavior: a review of research and theory. *Psychological bulletin*, 88(1):82–108, July 1980.

[58] M. Cohn. *User stories applied: For agile software development*. Addison-Wesley Professional, 2004.

[59] P. Collins, S. Shukla, and D. Redmiles. Activity theory and system design: A view from the trenches. *Computer Supported Cooperative Work (CSCW)*, 11(1):55–80, 2002.

[60] A. Cooper. *The inmates are running the asylum: Why high tech products drive us crazy and how to restore the sanity*. Pearson Higher Education, 2004.

[61] A. P. M. Coxon. *Sorting Data: Collection and Analysis*. SAGE Publications, 1999.

[62] A. Crystal and B. Ellington. Task analysis and human-computer interaction: approaches, techniques, and levels of analysis. In *Proceedings of the Tenth Americas Conference on Information Systems*, number August, pages 1–9, 2004.

[63] A. Cypher. Eager: Programming repetitive tasks by example. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, pages 33–39. ACM Press, 1991.

[64] D. Cypher, A. And Halbert. *Watch what I do: programming by demonstration*. The MIT press, 1993.

[65] M. Czerwinski, E. Horvitz, and S. Wilhite. A diary study of task switching and interruptions. In *Proceedings of the SIGCHI*. ACM Press, 2004.

[66] M. Czerwinski and E. Cutrell. Instant messaging and interruption: Influence of task type on performance. In *OZCHI 2000 conference*. ACM Press, 2000.

[67] M. Czerwinski and E. Horvitz. An investigation of memory for daily computing events. In *PEOPLE AND COMPUTERS*, pages 1–13. Springer, 2002.

[68] T. Davenport. *Human capital: What it is and why people invest it*. Jossey-Bass, CA, 1999.

[69] T. Davenport and S. Jarvenpaa. Improving knowledge work processes. *Sloan management review*, 37:53–66, 1996.

[70] T. Davenport and L. Prusak. *Working knowledge: How organizations manage what they know*. Harvard Business Press, Boston, 2000.

[71] F. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, 13(3):319, 1989.

[72] G. Davis. Affordances of ubiquitous computing and productivity in knowledge work. *Workshop on Ubiquitous Computing*, 2003.

[73] D. Diaper. Understanding task analysis for human-computer interaction. In D. Diaper and N. Stanton, editors, *The handbook of task analysis for human-computer interaction*, chapter 1, pages 5–47. Laurence Erlbaum Associates, London, 2004.

[74] J. F. Díaz-Morales, J. R. Cohen, and J. R. Ferrari. An integrated view of personality styles related to avoidant procrastination. *Personality and Individual Differences*, 45(6):554–558, October 2008.

[75] A. Dix, A. Katifori, A. Poggi, T. Catarci, Y. Ioannidis, G. Lepouras, and M. Mora. From Information to Interaction: in Pursuit of Task-centred Information Management. In *Proceedings of DELOS Conference*, 2007.

[76] S. Doeweling, B. Schmidt, and A. Goeb. A model for the design of interactive systems based on activity theory. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 539–548. ACM Press, 2012.

[77] A. Dragunov and T. Dietterich. TaskTracer: a desktop environment to support multi-tasking knowledge workers. In *Proceedings of the 10th international conference on Intelligent user interfaces*, pages 75–82. ACM Press, 2005.

[78] P. F. Drucker. Knowledge-Worker Productivity: The Biggest Challenge. *California Management Review*, 2(Winter):79–94, 1999.

[79] E. Duval. Attention Please ! Learning Analytics for Visualization and Recommendation. In *Proceedings of LAK11: 1st International Conference on Learning Analytics and Knowledge*. ACM Press, 2011.

[80] L. Eduardo and B. Daltrini. Activity Theory: a Framework to Software Requirements Elicitation. Citeseer, 2008.

[81] L. Efimova. Discovering the iceberg of knowledge work: A weblog case. *European Conference on Organisational Knowledge*, 2004.

[82] G. Einstein, M. McDaniel, and C. Williford. Forgetting of intentions in demanding situations is rapid. *Journal of Experimental Psychology*, 9(3):147–62, September 2003.

[83] C. Ellis, A. Rembert, K.-h. Kim, and J. Wainer. Beyond workflow mining. In *Springer*, volume 4102, pages 49–64. Springer, 2006.

[84] Y. Engeström. *Learning by expanding. An activity-theoretical approach to developmental research*. Orienta-Konsultit Oy, 1987.

[85] Y. Engeström. The future of activity theory: a rough draft. In *Learning and expanding with activity theory*, pages 303–328. Cambridge University Press Cambridge, 2009.

[86] B. A. F. Farhoomand and D. H. Drury. Managerial information overload. *Communications of the ACM*, 45(10):127 – 131, 2002.

[87] K. D. Fenstermacher and M. Ginsburg. A lightweight framework for cross-application user monitoring. *Computer*, 35(3):51–59, 2002.

[88] D. Ferreira, M. Zacarias, M. Malheiros, and P. Ferreira. Approaching process mining with sequence clustering: experiments and findings. *Business Process Management*, (1):360–374, 2007.

[89] D. Ferrucci and A. Lally. Uima: An architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10:327–348, 2004.

[90] J. Finlay, R. Beal, P. R. Beale, G. Abowd, T. Carey, R. Eberts, D. Eichmann, S. Grant, A. Halabieh, and J. Mcgrew. Neural networks and pattern recognition in human-computer interaction. *ACM SIGCHI Bulletin*, 25(April):25–35, 1993.

[91] M. Fjeld, K. Lauche, and M. Bichsel. Physical and Virtual Tools : Activity Theory Applied to the Design of Groupware. *Computer Supported Cooperative Work (CSCW)*, 11(1-2):153–180, 2002.

[92] S. Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, 2010.

[93] S. Friedenthal, A. Moore, and R. Steiner. *A practical guide to SysML: the systems modeling language*. Morgan Kaufmann, 2011.

[94] "Gallup San Jose State University". Managing Corporate Communications In The Information Age. 1999.

[95] A. Gangemi, S. Borgo, and C. Catenacci. Task taxonomies for knowledge content. *METOKIS Deliverable, D*, 2004.

[96] A. Gangemi, N. Guarino, and C. Masolo. Sweetening ontologies with DOLCE. In *Knowledge engineering and knowledge management: Ontologies and the semantic Web*, pages 223–233. Springer, 2002.

[97] E. Geisler. A typology of knowledge management: strategic groups and role behavior in organizations. *Journal of Knowledge Management*, 11(1):84–96, 2007.

[98] M. Ghoniem, J.-d. Fekete, and P. Castagliola. On the readability of graphs using node-link and matrix-based representations : a controlled experiment and statistical analysis. *Information Visualization*, 4(2):114–135, 2005.

[99] T. Gill and R. Hicks. Task complexity and informing science: A synthesis. *Informing Science*, 9, 2006.

[100] T. Gillie. What makes interruptions disruptive? A study of length, similarity, and complexity. *Psychological Research*, 50(4):243–250, 1989.

[101] V. González and G. Mark. Constant, constant, multi-tasking craziness: managing multiple working spheres. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, volume 6, pages 113–120. ACM Press, 2004.

[102] M. Granitzer, M. Kroll, C. Seifert, A. Rath, N. Weber, O. Dietzel, and S. Lindstaedt. Analysis of machine learning techniques for context extraction. In *Third International Conference on Digital Information Management, 2008.*, pages 233–240. IEEE Computer Society, 2008.

[103] T. Groza, S. Handschuh, K. Möller, G. Grimnes, L. Sauermann, E. Minack, M. Jazayeri, C. Mesnage, G. Reif, and R. Gudjónnsdóttir. The nepomuk project-on the way to the social semantic desktop. In S. Bechhofer, M. Hauswirth, J. Hoffmann, and M. Koubarakis, editors, *The Semantic: Web Research and Applications*. Springer, 2007.

[104] T. R. Gruber. A Translation Approach to Portable Ontology Specifications by A Translation Approach to Portable Ontology Specifications. *Knowledge acquisition*, 5(2):199–220, 1993.

[105] M. Grüninger and C. Menzel. Specification Language (PSL) Theory and Applications. *AI Magazine*, 24(3):63–74, 2003.

[106] C. Günther and W. van der Aalst. Mining activity clusters from low-level event logs. In *Beta, Research School for Operations Management and Logistics*, 2006.

[107] K. Gyllstrom, C. Soules, and A. Veitch. Activity put in context: identifying implicit task context within the user's document interaction. In *Proceedings of the second international symposium on Information interaction in context*, pages 51–56. ACM Press, 2008.

[108] W. Hacker. *General Work Psychology [Allgemeine Arbeitspsychologie - Psychische Regulation von Arbeitstätigkeiten]*. Verlag Hans Huber, Bern, Göttingen, Toronto, Seattle, 1998.

[109] T. Hädrich. *Situation-oriented Provision of Knowledge Services*. Dissertation, Martin Luther Universität Halle-Wittenberg, 2008.

[110] S. Han and G. Humphreys. Uniform connectedness and classical Gestalt principles of perceptual grouping. *Attention, Perception, & Psychophysics*, 61(4):661–674, 1999.

[111] H. Hasan and E. Gould. Activity-based Knowledge Management Systems. *Journal of Information & Knowledge Management*, 2(2):107–115, 2003.

[112] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2nd edition, 2009.

[113] M. Hatch and A. Cunliffe. *Organization theory: Modern, symbolic, and postmodern perspectives*. Oxford University Press, 2006.

[114] H. Heckhausen, P. Gollwitzer, and F. Weinert. *Jenseits des Rubikon: der Wille in den Humanwissenschaften*. Springer, 1987.

[115] G. Heinrich. Parameter estimation for text analysis. *Fraunhofer Technology Report*, 2009.

[116] D. Held. *Global transformations: Politics, economics and culture*. Stanford University Press, 1999.

[117] D. A. Henderson and S. K. Card. Rooms: the use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. *ACM Transactions on Graphics (TOG)*, 5(3):211–243, 1986.

[118] A. Hevner, S. March, J. Park, and S. Ram. Design science in information systems research. *MIS quarterly*, 28(1):75–105, 2004.

[119] D. Hilbert and D. Redmiles. Extracting usability information from user interface events. *ACM Computing Surveys (CSUR)*, 32(4):384–421, 2000.

[120] A. Hinze, K. Sachs, and A. Buchmann. Event-Based Applications and Enabling Technologies. In *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*. ACM Press, 2009.

[121] H. M. Hodgetts and D. M. Jones. Interruption of the Tower of London task: support for a goal-activation approach. *Journal of experimental psychology. General*, 135(1):103–15, February 2006.

[122] C. Holsapple and K. Jones. Exploring Secondary Activities of the Knowledge Chain. 12(1):3–31, 2005.

[123] C. W. Holsapple and K. Jones. Exploring Primary Activities of the Knowledge Chain. 11(3):155–174, 2004.

[124] H. Holz, O. Rostanin, A. Dengel, T. Suzuki, K. Maeda, and K. Kanasaki. Task-based process know-how reuse and proactive information delivery in TaskNavigator. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, page 531. ACM Press, 2006.

[125] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse. The Lumière project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 256–265. Morgan Kaufmann, 1998.

[126] E. Horvitz and A. Jacobs. Attention-sensitive alerting. In *Proceedings of the Fifteenth conference on Uncertainty and Artificial Intelligence*, pages 305–313. Morgan Kaufmann, 1999.

[127] L. Hosmer. Trust: The connecting link between organizational theory and philosophical ethics. *Academy of management Review*, 20(2):379–403, 1995.

[128] S. Houben. *Activity Theory Applied to the Desktop Metaphor*. PhD thesis, Hasselt University, 2011.

[129] Http://social.wakoopa.com/. Social wakoopa. http://social.wakoopa.com/.

[130] Http://www.rememberthemilk.com/. Remember the milk. http://www.rememberthemilk.com/.

[131] Http://www.rescuetime.com/. Rescue Time. http://www.rescuetime.com/.

[132] J. M. Hudson. "I'd Be Overwhelmed, But It's Just One More Thing to Do:" Availability and Interruption in Research Management. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, number 4, pages 97–104, 2002.

[133] IBM. IBM Connections Activities. http://www-01.ibm.com/software/lotus/products/connections/activities.html.

[134] S. T. Iqbal and B. P. Bailey. Oasis: A Framework for Linking Notification Delivery to the Perceptual Structure of Goal-Directed Tasks. *ACM Transactions on Computer-Human Interaction*, 17(4):1–28, December 2010.

[135] S. Iqbal. Understanding and developing models for detecting and differentiating breakpoints during interactive tasks. In *Proceedings of the SIGCHI conference on Human*, page 697, New York, New York, USA, 2007. ACM Press.

[136] ISO. Ergonomics of System Interaction - Part 210: Human-centered design for interactive systems, 2011.

[137] C. Jensen, H. Lonsdale, E. Wynn, and J. Cao. The life and times of files and information: a study of desktop provenance. In *Proceedings of the 28th CHI*, pages 767–776, 2010.

[138] J. Jin. Self-interruption on the computer: a typology of discretionary task interleaving. In *Proceedings of the 27th international conference on Computer-Human Interaction*, pages 1799–1808. ACM Press, 2009.

[139] S. L. Joslyn. *Individual differences in time pressured decision making*. PhD thesis, University of Washington, 1995.

[140] V. Kaptelinin and B. Nardi. *Acting with technology - Activity Theory and Interaction Design*. MIT Press, 2006.

[141] V. Kaptelinin. Activity theory: implications for human-computer interaction. In *Context and consciousness: Activity theory and human-computer interaction*, number 1, pages 103–116. MIT Press, 1996.

[142] V. Kaptelinin. The Activity Checklist : A Tool for Representing the "Space" of Context. *Magazine interactions*, 6(4):27–39, 1997.

[143] V. Kaptelinin. UMEA: translating interaction histories into project contexts. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, number 5, pages 353–360. ACM Press, 2003.

[144] V. Kaptelinin and B. A. Nardi. *Activity Theory in HCI - Fundamentals and Reflections*. Morgan & Claypool Publishers, synthesis edition, 2011.

[145] E. Kelloway. Knowledge work as organizational behavior. *International journal of management reviews*, 2(3):287–304, 2000.

[146] A. Kern. Lebensformen und epistemische Fähigkeiten. *Deutsche Zeitschrift für Philosophie*, 55(2):245–260, 2007.

[147] E. Kindler, V. Rubin, and W. Schäfer. Process mining and petri net synthesis. In *Business Process Management Workshops*, pages 105–116. Springer, 2006.

[148] S. Kuiper and J. Sklar. *Practicing Statistics: Guided Investigations for the Second Course*. Springer, 2012.

[149] K. Kuutti. Activity Theory as a potential framework for human- computer interaction research. In B. A. Nardi, editor, *Context and consciousness: Activity theory and human-computer interaction*, pages 17–44. MIT Press, 1995.

[150] M. Lamming, P. Brown, and K. Carter. The design of a human memory prosthesis. *The Computer Journal*, 37(3):1994, 1994.

[151] T. V. Landesberger and A. Kuijper. Visual analysis of large graphs. *of Euro-Graphics:*, 2010.

[152] A. Leontiev. *Activity, Consciousness, and Personality*. Prentice-Hall, Englewood Cliffs, NJ, 1978.

[153] A. Leontiev. *Problems in the Development of the Mind*. Progress Publishers, Moscow, 1981.

[154] A. N. Leontiev. *Activity and consciousness*. Progress Publishers, 1977.

[155] N. Lesh and O. Etzioni. A sound and fast goal recognizer. In *International Joint Conference on Artificial Intelligence*, volume 14, pages 1704–1710. Laurence Erlbaum Associates, 1995.

[156] Y. Li and N. Belkin. A faceted approach to conceptualizing tasks in information seeking. *Information Processing & Management*, 44(6):1822–1837, 2008.

[157] H. Lieberman. *Your wish is my command: Programming by example*. Morgan Kaufmann, 2001.

[158] Y. Lindell and B. Pinkas. Privacy Preserving Data Mining. In *Advances in Cryptology - CRYPTO 2000*. Springer, January 2000.

[159] H. Link, T. Lane, and J. Magliano. Models and Model Biases for Automatically Learning Task Switching Behavior. In *Foundations of augmented cognition*, pages 510–519. Lawrence Erlbaum Associates, 2005.

[160] H. Liu, S. Shah, and W. Jiang. On-line outlier detection and data cleaning. *Computers & Chemical Engineering*, 28(9):1635–1647, August 2004.

[161] R. Lokaiczyk, A. Faatz, A. Beckhaus, and M. Goertz. Enhancing Just-in-Time E-Learning Through Machine Learning on Desktop Context Sensors. In *Modeling and Using Context*, pages 330–341. Springer, 2007.

[162] R. Lokaiczyk, A. Faatz, and B. Schmidt. Useability versus Adaptation - Approaches to Higher Level Context Detection. In *Sixth International Workshop on Modeling and Reasoning in Context*, pages 25–37. CEUR-WS, 2010.

[163] R. Lokaiczyk and M. Goertz. Extending Low Level Context Events by Data Aggregation. In *Proceedings of I-KNOW '08 and I-MEDIA '08*, pages 118–125. know-center.tugraz.at, 2008.

[164] A. Lovrencic, M. Konecki, and T. Orehovacki. 1957-2007: 50 years of higher order programming languages. *Journal of Information and Organizational Sciences*, 33(1):79–150, 2009.

[165] J. Lu and S. Pan. Information at your fingertips: contextual IR in enterprise Email. In *Proceedings of the 16th international conference on Intelligent user interfaces*, pages 205–214. ACM Press, 2011.

[166] D. Luckham. *The power of events: an introduction to complex event processing in distributed enterprise systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[167] A. Luria. *Cognitive development, its cultural and social foundations*. Harvard University Press, Cambridge, 1976.

[168] F. Machlup. *The Production and Distribution of Knowledge in the United States*. Princeton University Press, 1973.

[169] P. Maglio, R. Barrett, C. Campbell, and T. Selker. SUITOR: An attentive information system. In *Proceedings of the 5th international conference on Intelligent user interfaces*, pages 169–176. ACM Press, 2000.

[170] M. Maguire. Context of Use within usability activities. *International Journal of Human-Computer Studies*, 55(4):453–483, October 2001.

[171] M. Maguire. Methods to support human-centred design. *International Journal of Human-Computer Studies*, 55(4):587–634, October 2001.

[172] J. Makolm. DYONIPOS: Proactive Knowledge Management. In *Proceedings of 21st Blede Conference eCollaboration: Overcoming Boundaries through Multi-Channel Interaction*, pages 475–482. Bled eConference, 2008.

[173] C. Manning, P. Raghavan, and H. Schütze. *An Introduction to Information Retrieval*. Number c. Cambridge University Press, 2009.

[174] G. Mark, V. M. Gonzalez, and J. Harris. No Task Left Behind? Examining the Nature of Fragmented Work. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 321–330. ACM Press, 2005.

[175] G. Mark and D. Gudith. The cost of interrupted work: more speed and stress. In *Proceedings of the twenty-sixth annual conference of CHI*, pages 107–110. ACM Press, 2008.

[176] M. Markus. Toward a theory of knowledge reuse: Types of knowledge reuse situations and factors in reuse success. *Journal of management information systems*, 18(1):57–93, 2001.

[177] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari, and I. Horrocks. WonderWeb Deliverable D18 Ontology Library. *Communities*, 2001.

[178] D. Maulsby. Inductive task modeling for user interface customization. In *Proceedings of the 2nd international conference on Intelligent user interfaces*, pages 236–240. ACM Press, 1997.

[179] D. C. Mcfarlane. Interruption of People in Human-Computer Interaction : A General Unifying Definition of Human Interruption and Taxonomy. 1997.

[180] D. C. Mcfarlane and K. A. Latorella. The Scope and Importance of Human Interruption in Human - Computer Interaction Design. *Human-Computer Interaction*, 17(1):1–61, 2002.

[181] D. Mcfarlane. *Interruption of people in human-computer interaction*. Dissertation, George Washington University, 1998.

[182] M. Memmel and A. Dengel. Sharing Contextualized Attention Metadata to Support Personalized Information Retrieval. *InternationalWorkshop on Contextualized Attention Metadata: Personalized Access to Digital Resources*, pages 19–26, 2007.

[183] Merriam-Webster.com. "interruption". http://www.merriam-webster.com/dictionary/interruption.

[184] Merriam-Webster.com. "work". http://www.merriam-webster.com/dictionary/work.

[185] "Microsoft Corporation". Windows Presentation Foundation. http://msdn.microsoft.com/en-us/library/ms754130.aspx.

[186] S. E. Middleton, N. R. Shadbolt, and D. C. D. E. Roure. Ontological User Profiling in Recommender Systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):54–88, 2004.

[187] G. a. Miller. The magical number seven, plus or minus two: some limits on our capacity for processing information. 1956. *Psychological review*, 101(2):343–52, April 1994.

[188] Y. Miyata and D. A. Norman. Psychological issues in support of multiple activities. In *User Centered Systems Design: New Perspectives on Human- Computer Interaction*, pages 265–284. L. Erlbaum Associates Inc., 1986.

[189] C. a. Monk, J. G. Trafton, and D. a. Boehm-Davis. The effect of interruption duration and demand on resuming suspended goals. *Journal of experimental psychology. Applied*, 14(4):299–313, December 2008.

[190] R. Morteo, V. Gonzalez, J. Favela, and G. Mark. Sphere juggler: fast context retrieval in support of working spheres. In *Proceedings of the Fifth Mexican International Conference in Computer Science, 2004. ENC 2004.*, pages 361–367. IEEE Computer Society, 2004.

[191] D. Mwanza. Where theory meets practice: A case for an Activity Theory based methodology to guide computer system design. In M. Hirose, editor, *Proceedings of INTERACT*, number February, pages 9–13. IOS Press, 2001.

[192] K. Myers and D. Wilkins. The Act Formalism, Version 2.2. *SRI International Artificial Intelligence Center Technical Report*, 1997.

[193] P. M. Nadkarni, L. Ohno-Machado, and W. W. Chapman. Natural language processing: an introduction. *Journal of the American Medical Informatics Association : JAMIA*, 18(5):544–51, 2011.

[194] R. Nair, S. Voida, and E. D. Mynatt. Frequency-based detection of task switches. In *Proceedings of the 19th British HCI Group Annual Conference*, volume 2, pages 94–99. British HCI Group, 2005.

[195] G. C. Neto, A. S. Gomes, J. Castro, and S. Sampaio. Integrating activity theory and organizational modeling for context of use analysis. In *Proceedings of the 2005 Latin American conference on Human-computer interaction - CLIHC '05*, pages 301–306, New York, New York, USA, 2005. ACM Press.

[196] D. Newtson. Attribution and the unit of perception of ongoing behavior. *Journal of Personality and Social Psychology*, 28(1):28–38, 1973.

[197] A. Noack. Energy models for graph clustering. *Journal of Graph Algorithms and Applications*, 11(2):453–480, 2007.

[198] D. Norman and S. Draper. *User centered system design; new perspectives on human-computer interaction*. L. Erlbaum Associates Inc., 1986.

[199] K. North. Produktive Wissensarbeit. In *5. Karlsruher Symposium für Wissensmanagement in Theorie und Praxis*, 2007.

[200] D. Oberle, S. Lamparter, S. Grimm, and D. Vrande. Towards Ontologies for Formalizing Modularization and Communication in Large Software Systems. In *Handbook on Ontologies*. Springer, 2006.

[201] B. O'Conaill. Timespace in the workplace: Dealing with interruptions. In *Conference companion on Human factors in computing systems*, pages 262–263. ACM Press, 1995.

[202] R. Oesterreich. *Handlungsregulation und Kontrolle*. Urban & Schwarzenberg, 1981.

[203] Office.microsoft.com. Create tasks and to-do items. http://office.microsoft.com/en-us/outlook-help/create-tasks-and-to-do-items-HA001229302.aspx.

[204] N. Oliver, G. Smith, C. Thakkar, and A. Surendran. SWISH: semantic analysis of window titles and switching history. In *Proceedings of the 11th international conference on Intelligent user interfaces*, pages 201–209. ACM Press, 2006.

[205] "Oxford Dictionaries". "coordination", 2012. http://oxforddictionaries.com.

[206] S. Palmer and I. Rock. Rethinking perceptual organization: The role of uniform connectedness. *Psychonomic Bulletin & Review*, 1(1):29—-55, 1994.

[207] H. Paulheim and F. Probst. A Formal Ontology on User Interfaces - Yet Another User Interface Description Language? In *2nd Workshop on Semantic Models for Adaptive Interactive Systems (SEMAIS)*. Springer, 2011.

[208] T. Pedersen, S. Patwardhan, and J. Michelizzi. Wordnet:: similarity-measuring the relatedness of concepts. In *Proceedings of the National Conference on Artificial Intelligence*, page 10241025. Association for the Advancement of Artificial Intelligence, 2004.

[209] B. Peintner and J. Dinger. Task assistant: Personalized task management for military environments. In *Proceedings of the Twenty-First Innovative Applications of Artificial Intelligence Conference*, pages 128–134. Association for the Advancement of Artificial Intelligence, 2009.

[210] A. Picot, H. Dietl, and E. Franck. *Organisation-Eine ökonomische Perspektive*. Schäffer-Poeschel, Stuttgart, 1999.

[211] S. Pinker. A Theory of Graph Comprehension. In *Artificial intelligence and the future of testing*, pages 73–126. Lawrence Erlbaum Associates, 1990.

[212] M. Polyanyi. *The tacit dimension*. Routledge & Kegan Paul, London, 1966.

[213] P. Pyöriä. The concept of knowledge work revisited. *Journal of Knowledge Management*, 9(3):116–127, 2005.

[214] A. S. Rath. *User Interaction Context - Studying and Enhancing Automatic User Task Detection on the Computer Desktop via an Ontology-based User Interaction Context Model*. PhD thesis, Graz University of Technology, 2010.

[215] A. S. Rath and S. N. Lindstaedt. UICO: An ontology-based user interaction context model for Automatic Task Detection on the Computer Desktop. In *Proceedings of the 1st Workshop on Context, Information and Ontologies*, number Ciao. ACM Press, 2009.

[216] A. Rath and N. Weber. Context-aware knowledge services. In *Proceedings of the EC-Tel. Lecture Notes in Computer Science, vol. 5192*, pages 234–244. Springer, 2008.

[217] T. Rattenbury and J. Canny. CAAD: an automatic task support system. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 696–706. ACM Press, 2007.

[218] T. L. Rattenbury and J. F. Canny. *An Activity Based Approach to Context-Aware Computing*. PhD thesis, University of California at Berkeley, 2008.

[219] RedHat. Drools Fusion. http://www.jboss.org/drools/drools-fusion.html.

[220] R. B. Reich. *The work of nations - Preparing Ourselves for 21st Century Capitalism*. Vintage Books New York, 1992.

[221] W. Reinhardt, B. Schmidt, P. Sloep, and H. Drachsler. Knowledge Worker Roles and Actions - Results of Two Empirical Studies. *Knowledge and Process Management*, 18(3):150–174, 2011.

[222] J. Rekimoto. Time-machine computing: a time-centric approach for the information environment. In *Proceedings of the 12th annual ACM symposium on User interface software and technology*, volume 1, pages 45–54, 1999.

[223] U. Riss and A. Rickayzen. Challenges for business process and task management. *Journal of Universal Knowledge Management*, 2:77–100, 2005.

[224] G. Robertson and M. V. Dantzich. The Task Gallery: a 3D window manager. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, volume 2, pages 494–501. ACM Press, 2000.

[225] G. Robertson and E. Horvitz. Scalable Fabric: flexible task management. In *Proceedings of the working conference on Advanced visual interfaces*, pages 85–89. ACM Press, 2004.

[226] C. Roda and J. Thomas. Attention aware systems: Theories, applications, and research agenda. *Computers in Human Behavior*, 22(4):557–587, 2006.

[227] C. Rosen. The myth of multitasking. *The New Atlantis*, (20):105–110, 2008.

[228] B. Rothrock, B. a. Myers, and S. H. Wang. Unified associative information storage and retrieval. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems extended abstracts on Human factors in computing systems*, page 1271, New York, New York, USA, 2006. ACM Press.

[229] S. L. Rubinshtein. The principle of creative self-activity (on the philosophical foundations of modern pedagogy) Originally published in 1922. *Voprosy Psikhologii*, 4:101–107, 1986.

[230] G. Salton, A. Wong, and C. S. Yang. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[231] D. D. Salvucci and N. a. Taatgen. Threaded cognition: an integrated theory of concurrent multitasking. *Psychological review*, 115(1):101–30, January 2008.

[232] D. D. Salvucci and N. A. Taatgen. *The multitasking mind*. Oxford University Press, 2010.

[233] L. Sauermann, L. Van Elst, and A. Dengel. Pimo-a framework for representing personal information models. In *Proceedings of I-Semantics*, volume 7, pages 270–277. Springer, 2007.

[234] M. Schief and B. Schmidt. Crossing the Chasm Between the Real World and Business Process Management. In *Business Process and Services Computing, Lecture Notes in Business Information Processing*, pages 280–287. Springer, 2010.

[235] A. Schmidt. Impact of context-awareness on the architecture of learning support systems. In C. Pahl, editor, *Architecture Solutions for E-learning Systems*, volume 2007. Idea-Group Publishing, 2007.

[236] B. Schmidt, S. Doeweling, and M. Mühlhäuser. Interaction History Visualization. In *Proceedings of the 30th annual international conference on the design of communication*. ACM Press, 2012.

[237] B. Schmidt and S. Döweling. Transparency - Activity Theory-Guided Support for Information Work. In *Context-Aware Adaptation of Service Front-Ends (CASFE 2012)*, 2012.

[238] B. Schmidt and E. Godehardt. Interaction Data Management. In *Knowlege-Based and Intelligent Information and Engineering Systems*. Springer, 2011.

[239] B. Schmidt, E. Godehardt, and B. Pantel. Visualizing the work process - Situation awareness for the knowledge worker. In *3rd IUI Workshop on Semantic Models for Adaptive Interactive Systems (SEMAIS 2012)*, 2012.

[240] B. Schmidt, E. Godehardt, and H. Paulheim. PASTREM: Proactive recommendations for information workers. In *Semantic Models for Adaptive Interactive Systems*. Springer, 2013.

[241] B. Schmidt, J. Kastl, T. Stoitsev, and M. Mühlhäuser. Hierarchical Task Instance Mining in Interaction Histories. In *Proceedings of the 29th annual international conference on Design of communication (SIGDOC)*. ACM Press, 2011.

[242] B. Schmidt, H. Paulheim, T. Stoitsev, and M. Mühlhäuser. Towards a Formalization of Individual Work Execution at Computer Workplaces. In *Conceptual Structures for Discovering Knowledge, Lecture Notes in Artificial Intelligence*, pages 270–284. Springer, 2011.

[243] B. Schmidt and U. Riss. Task patterns as means to experience sharing. In *Advances in Web Based Learning - ICWL 2009, Information Systems and Applications*, pages 353–362. Springer, 2009.

[244] B. Schmidt and M. Schief. Towards Agile Business Processes Based on the Internet of Things. In *Advanced Manufacturing and Sustainable Logistics, Lecture Notes in Business Information Processing*, pages 257–262. Springer, 2010.

[245] B. Schmidt, T. Stoitsev, and M. M. Towards Intention-Aware Systems. In *Proceedings of I-KNOW 2010, 10 th International Conference on Knowledge Management and Knowledge Technologies*, pages 355–363. know-center.tugraz.at, 2010.

[246] B. Schmidt, T. Stoitsev, and M. Mühlhäuser. Activity-Centric Support for Weakly-Structured Business Processes. In *Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems*, pages 251–260. ACM Press, 2010.

[247] B. Schmidt, T. Stoitsev, and M. Mühlhäuser. Analysis of Knowledge Work Execution at Computer Workplaces. In *Proceedings of 12th European Conference on Knowledge Management*. acpi, 2011.

[248] B. Schmidt, T. Stoitsev, and M. Mühlhäuser. Task Models for Intention-Aware Systems. *Journal of Universal Computer Science*, 17(10):1511–1526, 2011.

[249] U. Schultze and R. Boland. Knowledge management technology and the reproduction of knowledge work practices. *The Journal of Strategic Information Systems*, 9(2-3):193–212, September 2000.

[250] S. Schwarz. A context model for personal knowledge management. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, volume 3946, pages 18–33. Springer, 2005.

[251] a. J. Sellen, G. Louie, J. E. Harris, and a. J. Wilkins. What brings intentions to mind? An in situ study of prospective memory. *Memory (Hove, England)*, 5(4):483–507, July 1997.

[252] A. J. Sellen and R. H. R. Harper. *The Myth of the Paperless Office*. The MIT Press, 2003.

[253] S. Shapiro and M. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3):591–611, 1965.

[254] J. Shen. *Activity recognition in desktop environments*. PhD thesis, Oregon Statne University, 2009.

[255] J. Shen, E. Fitzhenry, and T. G. Dietterich. Discovering frequent work procedures from resource connections. In *Proceedingsc of the 13th international conference on Intelligent user interfaces - IUI '09*, pages 277–286. ACM Press, 2008.

[256] J. Shen, J. Irvine, X. Bao, M. Goodman, S. Kolibaba, A. Tran, F. Carl, B. Kirschner, S. Stumpf, and T. Dietterich. Detecting and correcting user activity switches: Algorithms and interfaces. In *Proceedings of the 13th international conference on Intelligent user interfaces*, pages 117–126. ACM Press, 2009.

[257] J. Shen, L. Li, and T. Dietterich. Real-time detection of task switches of desktop users. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, volume 7, pages 2868–2873. ACM Press, 2007.

[258] J. Shen, L. Li, T. G. Dietterich, and J. L. Herlocker. A hybrid learning system for recognizing user tasks from desktop activities and email messages. In *Proceedings of the 11th international conference on Intelligent user interfaces - IUI '06*, pages 86–92, New York, New York, USA, 2006. ACM Press.

[259] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE*, pages 336–343. IEEE Computer Society, 1996.

[260] H. Simon. *The sciences of the artificial*. MIT Press, 1996.

[261] N. Sinha, J. T. G. Brown, and R. H. S. Carpenter. Task switching as a two-stage decision process. *Journal of neurophysiology*, 95(5):3146–53, May 2006.

[262] D. Skyrme. *Knowledge Networking: Creating the Collaborative Enterprise*. Butterworth Heinemann, Boston, 1999.

[263] A. Smith. *An Inquiry into the Nature and Causes of the Wealth of Nations*. A. and C. Black, 1863.

[264] G. Smith and P. Baudisch. Groupbar: The taskbar evolved. In *OZCHI 2003 Conference for the Computer-Human Interaction Special Interest Group of the Human Factors Society of Australia*, pages 1–10. ACM Press, 2003.

[265] B. Sparrow, J. Liu, and D. M. Wegner. Google effects on memory: cognitive consequences of having information at our fingertips. *Science*, 333(6043):776–8, August 2011.

[266] C. Speier and J. Valacich. The effects of task interruption and information presentation on individual decision making. In *Proceedings of the eighteenth international conference on Information systems*, number Panko 1992, pages 21–36. ACM Press, 1997.

[267] C. Speier and J. Valacich. The influence of task interruption on individual decision making: An information overload perspective. *Decision Sciences*, 30(2):337–360, 1999.

[268] L. Sproull. The Nature of Managerial Attention. *Advances in information processing in organizations*, 1:9–27, 1984.

[269] M. Steyvers and T. Griffiths. Probabilistic Topic Models. In *Latent Semantic Analysis: A Road to Meaning*, volume 22, pages 1028–1040. Laurence Erlbaum Associates, July 2010.

[270] A. Sutcliffe. Scenario-based requirements engineering. In *Requirements engineering conference, 2003. Proceedings. 11th IEEE international*, pages 320–329. IEEE Computer Society, 2003.

[271] "The Apache Software Foundation". UIMA. http://uima.apache.org/.

[272] TheApacheSoftwareFoundation. Jena. http://jena.apache.org/.

[273] ThePixelLabs. Tasks.Show. http://code.msdn.microsoft.com/windowsdesktop/TasksShow-1bf01c8d.

[274] A. Toomela. Activity Theory Is a Dead End for Cultural-Historical Psychology. *CULTURE AND PSYCHOLOGY*, 6(3):353–364, 2000.

[275] A. Toomela. Commentary: Activity Theory is a Dead End for Methodological Thinking in Cultural Psychology Too. *CULTURE AND PSYCHOLOGY2*, 14(3):289–303, 2008.

[276] J. Trafton, E. M. Altmann, D. P. Brock, and F. E. Mintz. Preparing to resume an interrupted task: effects of prospective goal encoding and retrospective rehearsal. *International Journal of Human-Computer Studies*, 58(5):583–603, May 2003.

[277] E. Tulving and D. Thomson. Encoding specificity and retrieval processes in episodic memory. *Psychological review*, 80(5):352–373, 1973.

[278] A. Van Bergen. *Task Interruption*. North- Holland Publishing Co., Amsterdam, 1968.

[279] A. van Dam. Post-WIMP user interfaces. *Communications of the ACM*, 40(2):63–67, February 1997.

[280] W. van der Aalst, A. Hofstede, and M. Weske. Business process management: A survey. In *Lecture Notes in Computer Science*, pages 1–12. Springer, 2003.

[281] S. van Dongen. *Graph clustering by flow simulation*. Dissertation, Centre for Mathematics and Computer Science, 2000.

[282] G. Veiga and D. Ferreira. Understanding spaghetti models with sequence clustering for prom. In *Business Process Management Workshops*, pages 92–103. Springer, 2010.

[283] V. Venkatesh and F. D. Davis. A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies. *Management Science*, 46(2):186–204, February 2000.

[284] S. Viller. Human Factors in Requirements Engineering. 1997.

[285] S. Vinoski. CORBA: Integrating diverse applications within distributed heterogeneous environments. *Communications Magazine, IEEE*, 1997.

[286] J. Vlasblom and S. J. Wodak. Markov clustering versus affinity propagation for the partitioning of protein interaction graphs. *BMC bioinformatics*, 10:99, January 2009.

[287] S. Voida. Re-framing the desktop interface around the activities of knowledge work. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 211–220. ACM Press, 2008.

[288] S. Voida and E. Mynatt. Integrating virtual and physical context to support knowledge workers. *Pervasive Computing, IEEE*, 99(3):73–80, 2002.

[289] S. Voida, E. D. Mynatt, and B. Macintyre. Supporting Activity in Desktop and Ubiquitous Computing. In *Beyond the Desktop Metaphor:Designing Integrated Digital Work Environments*. MIT Press, 2007.

[290] M. Völkel. *Personal Knowledge Models with Semantic Technologies*. Ph.d. thesis, KIT, 2010.

[291] W. Volpert. *Handlungsstrukturanalyse als Beitrag zur Qualifikationsforschung*. Pahl-Rugenstein, 1974.

[292] L. S. Vygotsky. Genesis of higher mental functions. In *Psychological research in the USSR, vol.1*. Progress Publishers, Moscow, 1966.

[293] L. Vygotsky. *Mind in Society: The Development of Higher Psychological Processes*. University Press, USA, Harvard, 1978.

[294] L. Vygotsky. Instrumental method in psychology [Instrumentalnyj metod v psikhologii]. In A. Luria and M. Yaroshevsky, editors, *L. S. Vygotsky. CollectedWorks.V. 1. [L. S.Vygotsky. Sobranie Sochinenij.T. 1]*, pages 103–109. Moscow, 1982.

[295] C. Ware. *Information visualization: perception for design*. Morgan Kaufmann, 2004.

[296] M. Weber. Objectivity in Social Science and Social Policy. *The Methodology of the Social Sciences*, 1949.

[297] W. Weerkamp and M. de Rijke. Activity Prediction: A Twitter-based Exploration. In *SIGIR 2012 Workshop on Time-aware Information Access*, 2012.

[298] M. Wolpers, J. Najjar, K. Verbert, and E. Duval. Tracking actual usage: the attention metadata approach. *Journal of educational technology and society*, 10(3):106, 2007.

[299] N. Yorke-smith, S. Saadati, K. L. Myers, and D. N. Morley. Like an Intuitive and Courteous Butler: A Proactive Personal Agent for Task Management. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 337–344. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

[300] E. S. Yu. Social Modeling and i*. In *Conceptual Modeling: Foundations and Applications*, volume 5600, pages 99–121. Springer, 2009.

[301] J. Yu, B. Benatallah, and R. Saint-Paul. A framework for rapid integration of presentation components. In *Proceedings of the 16th international conference on World Wide Web*, number December, pages 923–932. ACM Press, 2007.

[302] J. M. Zacks and B. Tversky. Event structure in perception and conception. *Psychological bulletin*, 127(1):3–21, January 2001.

[303] B. Zeigarnik. Das Behalten erledigter und unerledigter Handlungen. *Psychologische Forschung*, (9):1–85, 1927.

[304] P. G. Zimbardo and R. J. Gerrig. *Psychology and life*. HarperCollins, New York, 1992.

[305] D. Zimmerman and C. West. Sex roles, interruptions and silences in conversation. *Language and sex: Difference and dominance*, pages 105–129, 1975.

[306] A. Zinnen, M. Goertz, and A. Faatz. Privacy Issues when Rolling out an E-Learning Solution. In *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pages 6394–6401. Ed/ITLib Digital Library, 2008.

All web pages cited in this work have been checked in April 2013. However, due to the dynamic nature of the WorldWideWeb, web pages can change.

List of Figures